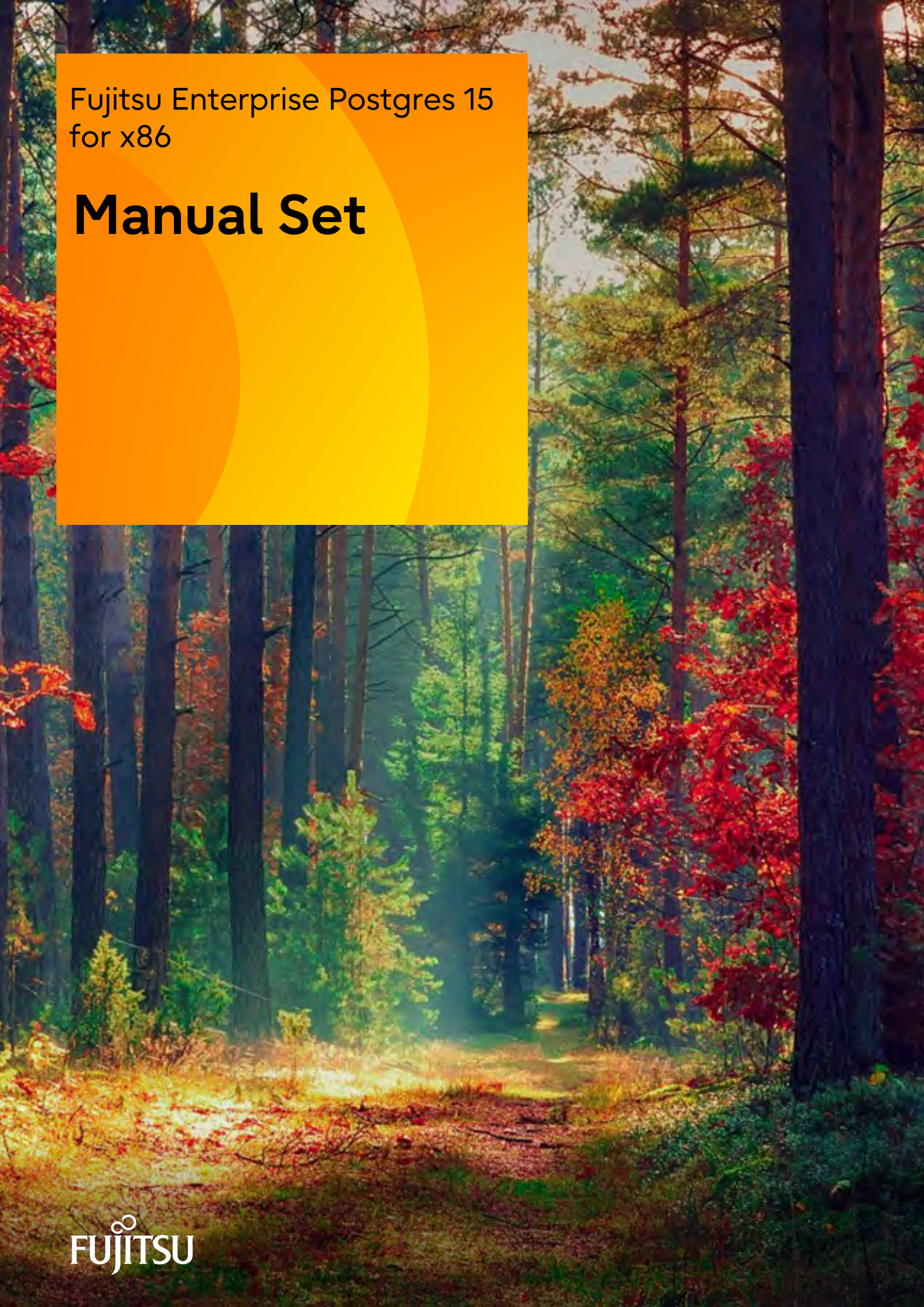
Fujitsu Enterprise Postgres 15
for x86

Manual Set



FUJITSU

Getting started

- Documentation Roadmap >
- Glossary >
- General Description >
- Release Notes >
- Program Updates>

DevSecOps

- Application Development Guide >
- Operation Guide >
- Security Operation Guide >
- Cluster Operation Guide >
- Connection Manager User's Guide >

Installation / Setup

- Server >
- Client >
- Server Assistant >
- Installation/uninstallation >
- Uninstall (middleware) >

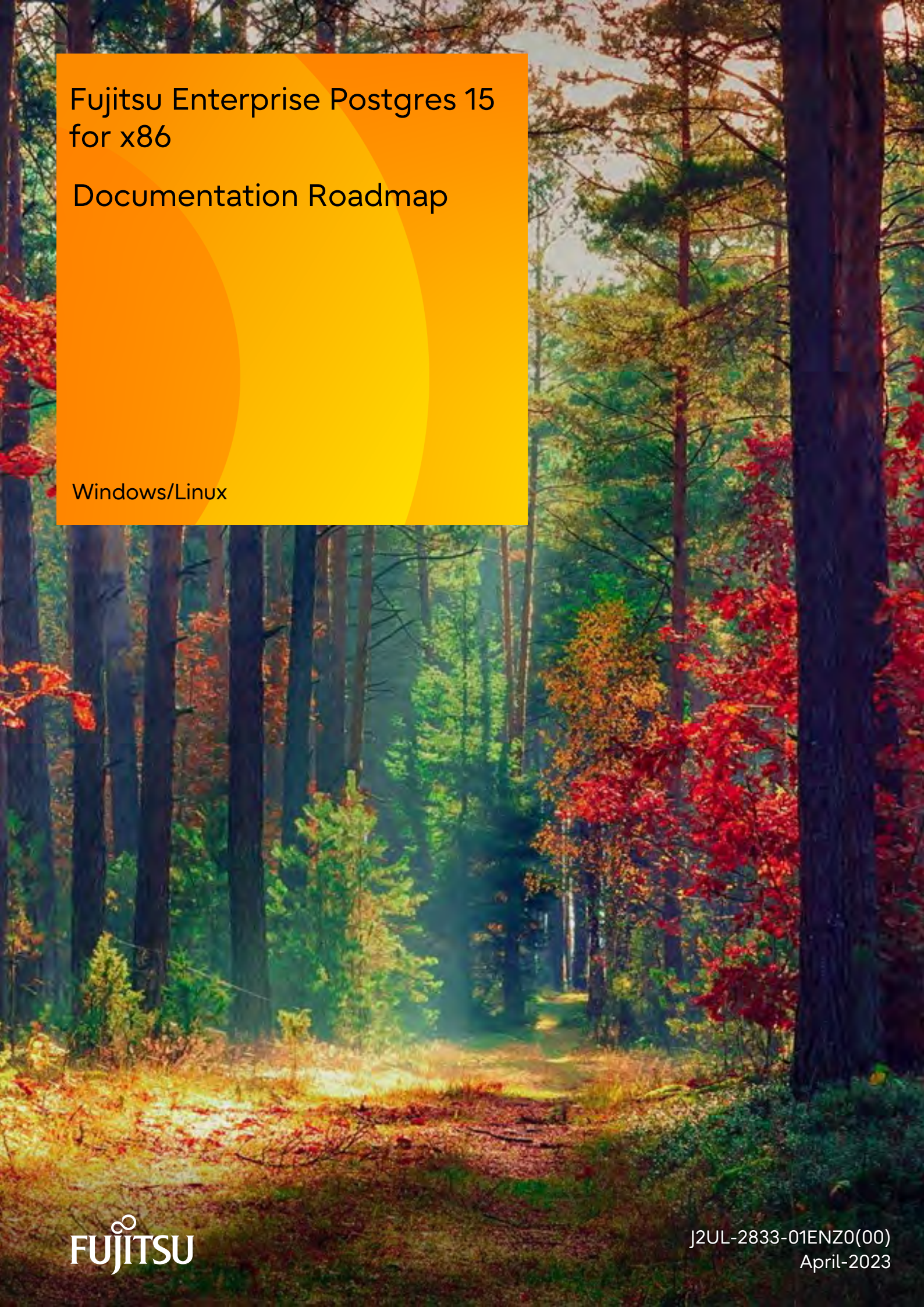
Reference

- Glossary >
- Reference Guide >
- Message Guide >
- FJQSS User's Guide >
- PostgreSQL 15.0 >

Fujitsu Enterprise Postgres 15 for x86

Documentation Roadmap

Windows/Linux



FUJITSU

J2UL-2833-01ENZ0(00)
April-2023

Preface

Purpose of this document

This document is intended for users of Fujitsu Enterprise Postgres, and explains how to read the manuals.

Structure of this document

The structure and content of this manual is shown below.

Chapter 1 How to Read the Manuals

This section explains the notational conventions in Fujitsu Enterprise Postgres manuals.

Chapter 2 Trademarks

This section explains the trademarks.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

```
Edition 1.0: April 2023
```

Copyright

Copyright 2015-2023 Fujitsu Limited

Contents

Chapter 1 How to Read the Manuals 1
1.1 Abbreviations of Manual Titles 1
1.2 System of Manuals and How to Use the Manuals 1
1.2.1 System of Manuals 1
1.2.2 Documentation Road Map 3
1.3 Notational Conventions in the Manuals 5
1.3.1 Platform-specific Information 5
1.3.2 Abbreviation of Product Names 5
1.3.3 Fujitsu Enterprise Postgres Conventions 7
1.3.3.1 Server 7
1.3.3.2 Client 7
1.3.4 Symbol Convention 7
1.4 Notes about Manuals 7

Chapter 2 Trademarks 8

Chapter 1 How to Read the Manuals

The Fujitsu Enterprise Postgres manuals use certain notational conventions and rules. Pay attention to these conventions and rules when reading the Fujitsu Enterprise Postgres manuals.

1.1 Abbreviations of Manual Titles

The following tables list abbreviations of the titles of manuals for Fujitsu Enterprise Postgres as they appear in the manuals.

Formal manual title	Abbreviation in Fujitsu Enterprise Postgres manuals
Fujitsu Enterprise Postgres Release Notes	Release Notes
Fujitsu Enterprise Postgres General Description	General Description
Fujitsu Enterprise Postgres Installation and Setup Guide for Server	Installation and Setup Guide for Server
Fujitsu Enterprise Postgres Installation and Setup Guide for Client	Installation and Setup Guide for Client
Fujitsu Enterprise Postgres Installation and Setup Guide for Server Assistant	Installation and Setup Guide for Server Assistant
Fujitsu Enterprise Postgres Operation Guide	Operation Guide
Fujitsu Enterprise Postgres Cluster Operation Guide (Database Multiplexing)	Cluster Operation Guide (Database Multiplexing)
Fujitsu Enterprise Postgres Cluster Operation Guide (PRIMECLUSTER)	Cluster Operation Guide (PRIMECLUSTER)
Fujitsu Enterprise Postgres Security Operation Guide	Security Operation Guide
Fujitsu Enterprise Postgres Application Development Guide	Application Development Guide
Fujitsu Enterprise Postgres Connection Manager User's Guide	Connection Manager User's Guide
Fujitsu Enterprise Postgres Reference	Reference
Fujitsu Enterprise Postgres Java API Reference	Java API Reference
Fujitsu Enterprise Postgres Npgsql API Reference	Npgsql API Reference
Fujitsu Enterprise Postgres Glossary	Glossary
Fujitsu Enterprise Postgres Messages	Messages
PostgreSQL *<x>* Documentation (*1)	PostgreSQL Documentation

*1: *<x>* indicates the version of PostgreSQL that Fujitsu Enterprise Postgres is based on. For the version, refer to "PostgreSQL Version Used for Fujitsu Enterprise Postgres" in the "Installation and Setup Guide for Server".

1.2 System of Manuals and How to Use the Manuals

This section describes the system of manuals for Fujitsu Enterprise Postgres.

1.2.1 System of Manuals

Fujitsu Enterprise Postgres manuals

The table below shows the manuals on Fujitsu Enterprise Postgres.

Use/Purpose	Manual title	Content	When to read
Deciding whether to upgrade the product.	Release Notes	Overview of upgraded features and incompatibility information.	When learning about features upgraded from earlier versions and incompatibility information.
Acquiring an overview of the product and the basic information required for work and operation.	General Description	Description of all available functions associated with each intended purpose or use, and screenshots of operations.	When learning basic information and restrictions that system engineers and operators must know to actually operate the product.
Installing and setting up Fujitsu Enterprise Postgres correctly to enable its use.	Installation and Setup Guide for Server	Procedure for installing and setting up Fujitsu Enterprise Postgres.	When installing and setting up Fujitsu Enterprise Postgres.
Installing the Fujitsu Enterprise Postgres client function correctly to enable its use.	Installation and Setup Guide for Client	Installing the Fujitsu Enterprise Postgres client function.	When installing the Fujitsu Enterprise Postgres client function.
Installing and setting up the Fujitsu Enterprise Postgres Server Assistant.	Installation and Setup Guide for Server Assistant	Procedure for installing and setting up the Fujitsu Enterprise Postgres Server Assistant.	When installing and setting up the Fujitsu Enterprise Postgres Server Assistant.
Operating and managing Fujitsu Enterprise Postgres.	Operation Guide	Description of the tasks required in Fujitsu Enterprise Postgres management and operation.	When learning how to operate and manage the databases.
Performing switchover using database multiplexing mode.	Cluster Operation Guide (Database Multiplexing)	Description of the tasks required for database multiplexing operation.	When using database multiplexing mode to create operating environment for switchover and perform it.
Performing failover using PRIMECLUSTER.	Cluster Operation Guide (PRIMECLUSTER)	Description of how to set up failover and perform operations using PRIMECLUSTER.	When performing failover using PRIMECLUSTER.
Performing security operation.	Security Operation Guide	Description of the tasks required for security operations.	When using security features and performing security operation in Fujitsu Enterprise Postgres.
Applications using the interface provided by Fujitsu Enterprise Postgres.	Application Development Guide	Procedure for creating an application using embedded SQL, JDBC driver, ODBC driver, and .NET Data Provider.	When developing an application using the interface provided by Fujitsu Enterprise Postgres.
Performing high availability system using	Connection Manager User's Guide	Description of the features, setup, and	When using Connection Manager features by Fujitsu Enterprise Postgres.

Use/Purpose	Manual title	Content	When to read
the Connection Manager feature.		usage of Connection Manager.	
Usage of Fujitsu Enterprise Postgres commands.	Reference	Description of the Fujitsu Enterprise Postgres commands expanded on from PostgreSQL.	When learning Fujitsu Enterprise Postgres command functions, options, and examples of use.
Learning the syntax of the JDBC API.	Java API Reference	Description of the syntax of the JDBC API.	When learning the syntax of the JDBC API.
Learning the syntax of the Npgsql API.	Npgsql API Reference	Description of the syntax of the Npgsql API.	When learning the syntax of the Npgsql API.
Learning the meaning of the terms of Fujitsu Enterprise Postgres.	Glossary	Description of the terms used in the Fujitsu Enterprise Postgres manuals.	When checking the meaning of terms used in the Fujitsu Enterprise Postgres manuals.
Referring to messages from Fujitsu Enterprise Postgres and taking measures for them.	Messages	Description of each message and description of any measures to be taken for it.	When finding out the specific measures for dealing with messages from Fujitsu Enterprise Postgres.

PostgreSQL manual

The table below shows the manual on PostgreSQL-compatible features.

Use/Purpose	Manual title	Content	When to read
Learning about PostgreSQL features.	PostgreSQL Documentation	Official PostgreSQL documentation. Explains all features officially supported by the relevant version of PostgreSQL.	When learning how to use PostgreSQL.

1.2.2 Documentation Road Map

This section provides a documentation roadmap, broken down by user role.

Database administrator

The database administrator is a user who performs Fujitsu Enterprise Postgres installation and setup, and who operates and monitors the database.

Refer to the manuals in the table below, according to purpose:

	Purpose	Manual name
Required reading	To learn about upgraded features and incompatibility information	Release Notes

	Purpose	Manual name
	To read an overview of the software	General Description
	To perform installation and setup	Installation and Setup Guide for Server
		Cluster Operation Guide (Database Multiplexing)
		Cluster Operation Guide (PRIMECLUSTER)
		Connection Manager User's Guide
	To install the Server Assistant	Installation and Setup Guide for Server Assistant
	To operate and monitor	Operation Guide
		Security Operation Guide
		Cluster Operation Guide (Database Multiplexing)
		Cluster Operation Guide (PRIMECLUSTER)
		Reference
	Using Connection Manager features	Connection Manager User's Guide
	Reference	Messages
		Glossary
Refer to as required	To learn about PostgreSQL features	PostgreSQL Documentation

Application developer

The application developer is a user who defines the database and develops applications.

Refer to the manuals in the table below, according to purpose:

	Purpose	Manual name
Required reading	To learn about upgraded features and incompatibility information	Release Notes
	To read an overview of the software	General Description
	To perform installation and setup	Installation and Setup Guide for Client
	To define a database	Operation Guide
	To develop applications	Application Development Guide
		Java API Reference
		Npgsql API Reference
	Using Connection Manager features	Connection Manager User's Guide

	Purpose	Manual name
	Reference	Messages
		Glossary
Refer to as required	To learn about PostgreSQL features	PostgreSQL Documentation

1.3 Notational Conventions in the Manuals

Manual titles and product names in the manual are abbreviated.

This section explains the notational conventions for abbreviations and platform-specific information in the manuals.

1.3.1 Platform-specific Information

Even manuals whose title has a platform name contains content common to all the platforms supported by Fujitsu Enterprise Postgres. In such cases, the platform-specific information is marked as shown below. Refer to only the necessary information.

Linux

Indicates content concerning Linux.

Windows(R)

Indicates content concerning Windows(R).

1.3.2 Abbreviation of Product Names

The following table lists abbreviations of the names of products related to Fujitsu Enterprise Postgres as they appear in the manuals.

Formal name	Abbreviation
Oracle Solaris 10 and Oracle Solaris 11	Solaris
Red Hat(R) Enterprise Linux(R) 7, Red Hat(R) Enterprise Linux(R) 8, Red Hat(R) Enterprise Linux(R) 9, SUSE Linux Enterprise Server 12 and SUSE Linux Enterprise Server 15	Linux
Red Hat(R) Enterprise Linux(R) 7	RHEL7
Red Hat(R) Enterprise Linux(R) 8	RHEL8
Red Hat(R) Enterprise Linux(R) 9	RHEL9
SUSE Linux Enterprise Server 12	SLES 12
SUSE Linux Enterprise Server 15	SLES 15
Windows(R) 10 Home, Windows(R) 10 Education, Windows(R) 10 Pro and Windows(R) 10 Enterprise	Windows(R) 10
Windows(R) 11 Home, Windows(R) 11 Education, Windows(R) 11 Pro and Windows(R) 11 Enterprise	Windows(R) 11

Formal name	Abbreviation
Microsoft(R) Windows Server(R) 2016 Datacenter, Microsoft(R) Windows Server(R) 2016 Standard and Microsoft(R) Windows Server(R) 2016 Essentials	Windows Server(R) 2016
Microsoft(R) Windows Server(R) 2019 Datacenter, Microsoft(R) Windows Server(R) 2019 Standard and Microsoft(R) Windows Server(R) 2019 Essentials	Windows Server(R) 2019
Microsoft(R) Windows Server(R) 2022 Datacenter, Microsoft(R) Windows Server(R) 2022 Standard and Microsoft(R) Windows Server(R) 2022 Essentials	Windows Server(R) 2022
Windows(R) 10 Home, Windows(R) 10 Education, Windows(R) 10 Pro, Windows(R) 10 Enterprise, Windows(R) 11 Home, Windows(R) 11 Education, Windows(R) 11 Pro, Windows(R) 11 Enterprise, Microsoft(R) Windows Server(R) 2016 Datacenter, Microsoft(R) Windows Server(R) 2016 Standard, Microsoft(R) Windows Server(R) 2016 Essentials, Microsoft(R) Windows Server(R) 2019 Datacenter, Microsoft(R) Windows Server(R) 2019 Standard, Microsoft(R) Windows Server(R) 2019 Essentials, Microsoft(R) Windows Server(R) 2022 Datacenter, Microsoft(R) Windows Server(R) 2022 Standard and Microsoft(R) Windows Server(R) 2022 Essentials	Windows(R)
Microsoft(R) Edge	Edge
Java Naming and Directory Interface	JNDI
Java(TM) 2 SDK, Standard Edition, Java(TM) 2 Platform, Enterprise Edition, Java(TM) Platform, Standard Edition and Java(TM) Development Kit	JDK
Java(TM) 2 Runtime Environment, Standard Edition and Java(TM) Runtime Environment	JRE
Microsoft(R) Visual Basic(R) for Applications	VBA
Microsoft(R) Visual Basic	Visual Basic
Microsoft(R) Visual Basic.NET	Visual Basic.NET
Microsoft(R) Visual Studio	Visual Studio
Microsoft(R) Visual Studio.NET	Visual Studio.NET
Microsoft .NET Framework	.NET Framework or.NET
Interstage Application Server Enterprise Edition, Interstage Application Server Standard-J Edition and Interstage Web Server	Interstage Application Server
Fujitsu Enterprise Postgres Advanced Edition (64bit)	AE or Fujitsu Enterprise Postgres Advanced Edition
Fujitsu Enterprise Postgres Standard Edition (64bit)	SE or Fujitsu Enterprise Postgres Standard Edition

Remarks: The symbols (R) and (TM) may be omitted in this manual.

1.3.3 Fujitsu Enterprise Postgres Conventions

The naming conventions for the Fujitsu Enterprise Postgres product names and functions used in the Fujitsu Enterprise Postgres manuals are shown below.

1.3.3.1 Server

The names used in the manuals in explanations regarding Fujitsu Enterprise Postgres functions are shown below.

Product name	Name used in manuals
Fujitsu Enterprise Postgres Advanced Edition (64bit) and Fujitsu Enterprise Postgres Standard Edition (64bit)	64-bit product

1.3.3.2 Client

The names used in the manuals in explanations regarding Fujitsu Enterprise Postgres client functions are shown below.

Product name	Name used in manuals
Fujitsu Enterprise Postgres Client (64bit)	64-bit product
Fujitsu Enterprise Postgres Client (32bit)	32-bit product

1.3.4 Symbol Convention

The symbols shown below are used in the manuals.

Symbol	Meaning
[ ]	These symbols indicate characters displayed in a window or dialog box or keyboard keys. Examples: [Setting] dialogue box, [File] menu, [Item name], [OK] button, [Enter] key.

1.4 Notes about Manuals

This section contains notes about the Fujitsu Enterprise Postgres operating environments and manuals.

- Images in figures

  The Fujitsu Enterprise Postgres manuals contain figures showing printouts for Fujitsu Enterprise Postgres to provide the reader an idea of what the printouts look like, but since the figures are only examples, they are incomplete.

- Explanatory examples

  Most of the examples of databases in the Fujitsu Enterprise Postgres manuals are modeled after inventory control databases of retail stores. The design and contents of the databases in the examples are fictitious and do not represent any real database.

- UNIX release version number

  This system conforms to UNIX System V Rel4.2MP.

Chapter 2 Trademarks

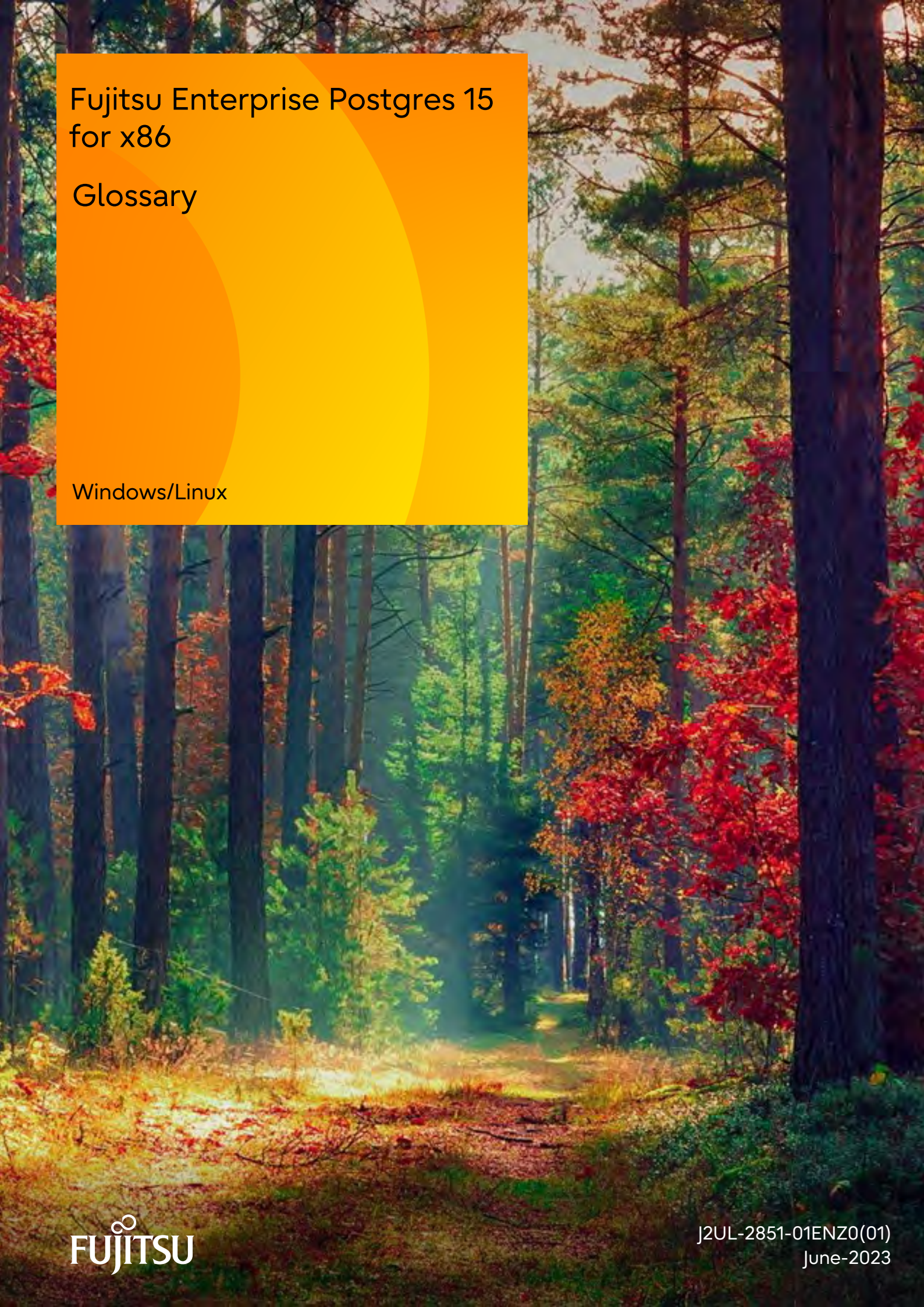
- Internet Information Services, Microsoft, MS, MS-DOS, Windows, Windows Server, Visual Basic, Visual Studio and Microsoft .NET Framework are registered trademarks or trademarks of Microsoft Corporation in the U.S. and/or other countries.
- Oracle and Java are registered trademarks of Oracle Corporation and its subsidiaries and affiliated companies in the U.S. and other countries. Product and company names mentioned in this manual are the trademarks or registered trademarks of their respective owners.
- Linux(R) is a registered trademark of Linus Torvalds in the U.S. and other countries.
- Red Hat, RPM, and all Red Hat-based trademarks and logos are registered trademarks or trademarks of Red Hat, Inc. in the U.S. and other countries.
- SUSE and the SUSE logo are registered trademarks of SUSE LLC in the United States and other countries.
- UNIX is a registered trademark of Open Group in the U.S. and other countries.
- Interstage, Systemwalker, PRIMECLUSTER, and Fujitsu Enterprise Postgres are trademarks or registered trademarks of Fujitsu Limited.

Other product and company names mentioned in this manual are the trademarks or registered trademarks of their respective owners.

Fujitsu Enterprise Postgres 15 for x86

Glossary

Windows/Linux



FUJITSU

J2UL-2851-01ENZ0(01)
June-2023

Preface

Purpose of this document

This document explains Fujitsu Enterprise Postgres terminology.

Intended readers

This document is aimed at all users of Fujitsu Enterprise Postgres.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

```
Edition 1.1: June 2023
Edition 1.0: April 2023
```

Copyright

Copyright 2015-2023 Fujitsu Limited

Revision History

Revision	Location	Version
Feature names have been improved. Before modification : Confidential group After modification : Confidentiality group Before modification : Confidential group role After modification : Confidentiality group role Before modification : Confidential level After modification : Confidentiality level Before modification : Confidential management role After modification : Confidentiality management role Before modification : Confidential management support After modification : Confidentiality management Before modification : Confidential matrix After modification : Confidentiality matrix Before modification : Confidential object After modification : Confidentiality object Before modification : Confidential privilege After modification : Confidentiality privilege	Glossary	Edition 1.1

Contents

Glossary..... 1

Index.....5

Glossary

Arbitration command

A user exit (user command) called when an abnormality is detected using operating system/server heartbeat monitoring in database multiplexing mode.

Arbitration server

A dedicated server on which the Server Assistant program is installed.

Archive log

Contains the history of updates made to the database, and is used during recovery.

Backup data storage destination

The directory that stores the backup data.

Client command

A command that is executed from the client machine and used. Also known as a client application.

Confidentiality group

An object that indicates which confidentiality level roles in confidentiality group can acdess.

Confidentiality group role

The targets of GRANT and REVOKE statements executed internally by the confidentiality management feature.

Confidentiality level

A group of data with the same degree of confidentiality.

Confidentiality object

A collection of data classified by confidentiality level, representing an object in the database.

Confidentiality privilege

The privileges indicate how a confidentiality group can access confidentiality objects registered at a certain confidentiality level.

Confidentiality management role

A role that manages confidentiality matrices.

Confidentiality Management

A feature that supports database users in setting appropriate privileges for each database resource.

Confidentiality matrix

A matrix of confidentiality levels and confidentiality groups.

Connection Manager

The replication operation to continue without knowing where the application is connected.

The Connection Manager feature improves the availability.

Data storage destination

The directory that stores the database clusters.

Database cluster

The database storage area on the database storage disk. Database clusters are a collection of databases managed by an instance.

Data masking

A feature that can change the returned data for queries generated by applications, to prevent exposing actual data.

Database multiplexing

Mechanism in which a database is made redundant on multiple servers, by transferring transaction logs (WAL) via the network to enable application jobs to be continued.

Database superuser

A user defined in the database with access privileges for all database objects.

Encoding

Indicates the character set.

Fencing

A process that isolates a database server with an unstable status from the cluster system in database multiplexing mode. This process is implemented as a fencing command.

Fencing command

A user exit (user command) that implements fencing in database multiplexing mode.

Global Meta Cache

The Global Meta Cache feature cache the informations about system catalogs information (catalog meta cache) in shared memory. The catalog meta cache on shared memory is called the Global Meta Cache (GMC).

Instance

A series of server processes for managing database clusters.

Instance administrator

The OS user account that owns the database cluster files and operates the database server processes.

Instance name

Indicates the instance name.

Key management system

A system that manages data encryption keys when using the transparent data encryption feature.

Local Meta Cache Limit

The ability to limit the size by removing the Local Meta Cache that has not been accessed for a long time.

Local Meta Cache is a meta cache (system catalog and table definition information) held in local memory.

Masking policy

A method of changing data under specific conditions when it is returned for a query from an application. You can configure masking target, masking type, masking condition and masking format.

Mirrored transaction log

The log that mirrors the transaction log at the backup data storage destination.

Mirroring Controller arbitration process

A process that performs arbitration and fencing on the arbitration server.

Mirroring Controller monitoring process

A process that performs heartbeat monitoring of the Mirroring Controller process. If the Mirroring Controller process returns no response or is down, the Mirroring Controller monitoring process is restarted automatically.

Mirroring Controller process

A process that performs operating system/server and process heartbeat monitoring and disk abnormality monitoring between database servers. Additionally, the process issues arbitration requests to the arbitration server and executes arbitration commands.

Pgpool-II connection pooling

The connection pooling feature of Pgpool-II supported by Fujitsu Enterprise Postgres.

This feature maintains the connection established with the database server and reuses that connection each time a new connection with the same properties (user name, database, and protocol version) arrives. By reducing the connection overhead for the database server, throughput of the whole system is improved.

Pgpool-II failover

The automatic failover feature of Pgpool-II supported by Fujitsu Enterprise Postgres.

If any of the database servers crashes or can no longer be reached, this feature disconnects the server and continues operation on the remaining servers. The streaming replication feature of PostgreSQL is combined with Pgpool-II to achieve a high-availability system.

Pgpool-II load balancing

The load balancing feature of Pgpool-II supported by Fujitsu Enterprise Postgres.

This feature distributes reference queries to multiple database servers, improving throughput of the whole system. The database multiplexing feature or PostgreSQL streaming replication feature is combined with Pgpool-II to reduce the load on the database server.

Pgpool-II server

A server for using the failover, connection pooling, and load balancing features of Pgpool-II. It is a dedicated server that has a server program installed for using these features.

Primary server

The server that processes the main database jobs during multiplexed database operation.

Server Assistant

A feature that objectively determines the status of database servers as a third party, and if necessary, isolates affected databases if the database servers are unable to accurately ascertain their mutual statuses in database multiplexing mode, such as due to a network error between database servers, or server instability.

Server Assistant program

A program to be installed on the arbitration server.

Server command

A command used on the database server. Also known as a server application.

Standby server

A server that generates a replicated database synchronized with the primary server, and that can run as an alternative server in case the primary server fails during multiplexed database operation.

State transition command

A user exit (user command) called when Mirroring Controller performs a state transition of a database server in database multiplexing mode. State transition commands include the post-switch command, pre-detach command, and post-attach command.

Transaction log

Contains the history of updates made to the database by transactions. Also known as the WAL (Write-Ahead Log).

Transaction log storage destination

The directory that stores the transaction log.

VCI (Vertical Clustered Index)

An index with columnar data structure suitable for aggregation.

WAL (Write-Ahead Log)

Has the same meaning as 'transaction log'.

WebAdmin program

A GUI-based program installed on a database server or a dedicated WebAdmin server, used to manage database instances.

WebAdmin server

By using the WebAdmin program on a different server to the database server, instances on multiple database servers can be managed from a dedicated WebAdmin server on which the WebAdmin program is installed.

Index

[A]

Arbitration command....1
Arbitration server....1
Archive log....1

[B]

Backup data storage destination....1

[C]

Client command....1
Confidentiality group....1
Confidentiality group role....1
Confidentiality level....1
Confidentiality Management....1
Confidentiality management role....1
Confidentiality matrix....1
Confidentiality object....1
Confidentiality privilege....1
Connection Manager....1

[D]

Database cluster....2
Database multiplexing....2
Database superuser....2
Data masking....2
Data storage destination....1

[E]

Encoding....2

[F]

Fencing....2
Fencing command....2

[G]

Global Meta Cache....2

[I]

Instance....2
Instance administrator....2
Instance name....2

[K]

Key management system....2

[L]

Local Meta Cache Limit....2

[M]

Masking policy....2
Mirrored transaction log....2
Mirroring Controller arbitration process....3
Mirroring Controller monitoring process....3
Mirroring Controller process....3

[P]

Pgpool-II connection pooling....3
Pgpool-II failover....3
Pgpool-II load balancing....3
Pgpool-II server....3
Primary server....3

[S]

Server Assistant....3
Server Assistant program....3
Server command....3
Standby server....3
State transition command....4

[T]

Transaction log....4
Transaction log storage destination....4

[V]

VCI (Vertical Clustered Index)....4

[W]

WAL(Write-Ahead Log)....4
WebAdmin program....4
WebAdmin server....4

FUJITSU

Fujitsu
Enterprise Postgres 15

General Description

Windows/Linux

J2UL-2836-01ENZ0(03)
March 2025

Preface

Purpose of this document

This document explains the Fujitsu Enterprise Postgres concepts to those who are to operate databases using it.

This document explains the features of Fujitsu Enterprise Postgres.

Intended readers

This document is intended for people who are:

- Considering installing Fujitsu Enterprise Postgres
- Using Fujitsu Enterprise Postgres for the first time
- Wanting to learn about the concept of Fujitsu Enterprise Postgres
- Wanting to see a functional overview of Fujitsu Enterprise Postgres

Readers of this document are also assumed to have general knowledge of:

- Computers
- Jobs
- Linux
- Windows(R)

Structure of this document

This document is structured as follows:

Chapter 1 Fujitsu Enterprise Postgres Basics

Explains the features of Fujitsu Enterprise Postgres.

Appendix A List of Features

Lists the main features provided by Fujitsu Enterprise Postgres.

Appendix B OSS Supported by Fujitsu Enterprise Postgres

Explains the OSS supported by Fujitsu Enterprise Postgres.

Appendix C Features that can be Used on Servers Other than the Database Server

Explains features that can be used on servers other than the database server.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

```
Edition 1.3: March  2025
Edition 1.2: March  2024
Edition 1.1: June  2023
Edition 1.0: April  2023
```

Copyright

Copyright 2015-2025 Fujitsu Limited

Revision History

Revision	Location	Version
Database Monitoring with Datasentinel to the feature list.	Appendix A List of Features	Edition 1.3
PostGIS has been added as a supported OSS.	Appendix B OSS Supported by Fujitsu Enterprise Postgres	Edition 1.2
Feature names have been improved. Before modification : Confidential group After modification : Confidentiality group Before modification : Confidential level After modification : Confidentiality level Before modification : Confidential management support After modification : Confidentiality management	Chapter 1 Fujitsu Enterprise Postgres Basics 1.9 Management of Access Control by Confidentiality Management Appendix A List of Features	Edition 1.1

Contents

Chapter 1 Fujitsu Enterprise Postgres Basics 1
1.1 Flexible Database Recovery 2
1.2 Simple GUI-Based Installation and Operation Management 4
1.3 High Reliability with Database Multiplexing 4
1.4 High Reliability Using Failover Integrated with the Cluster Software 6
1.5 Seamless Migration from Oracle Databases 7
1.6 Storage Data Protection Using Transparent Data Encryption 7
1.7 Data Masking for Improved Security 8
1.8 Security Enhancement Using Audit Logs 9
1.9 Management of Access Control by Confidentiality Management 9
1.10 Enhanced Query Plan Stability 10
1.11 Increased Aggregation Performance Using the In-memory Feature 11
1.12 High-Speed Data Load 12
1.13 High availability by using Connection Manager 13
1.14 Memory Usage Reduce with Meta cache Reduction and Limit 14
1.14.1 Memory Usage Reduction Using Global Meta Cache 14
1.14.2 Memory Usage Reduction Using Local Meta Cache Limit 15

Appendix A List of Features 17

Appendix B OSS Supported by Fujitsu Enterprise Postgres 18

Appendix C Features that can be Used on Servers Other than the Database Server 20
C.1 WebAdmin 20
C.2 Server Assistant 20
C.3 Failover, Connection Pooling, and Load Balancing Features of Pgpool-II 20

Index 23

Chapter 1 Fujitsu Enterprise Postgres Basics

Fujitsu Enterprise Postgres maintains the operating methods, interfaces for application development and SQL compatibility of PostgreSQL, while providing expanded features for enhanced reliability and operability.

This chapter explains the functionality extended by Fujitsu Enterprise Postgres.

Refer to "Appendix A List of Features" for feature differences between editions.

Additionally, Fujitsu Enterprise Postgres supports various open source software (OSS). Refer to "Appendix B OSS Supported by Fujitsu Enterprise Postgres" for information on OSS supported by Fujitsu Enterprise Postgres.

Fujitsu Enterprise Postgres has the following features:

- Flexible database recovery
  Not only does Fujitsu Enterprise Postgres recover data to its most recent form when a failure occurs, which is essential for databases, but it can also recover to any point in time. Additionally, backup/recovery can be performed using any copy technology.

- Simple GUI-based installation and operation management
  Fujitsu Enterprise Postgres uses GUI to simplify cumbersome database operations, and allows databases to be used intuitively.

- High reliability by using database multiplexing
  Database multiplexing protects important data and enables highly reliable database operation.

L

- High reliability by using the failover feature integrated with the cluster software
  Fujitsu Enterprise Postgres links with PRIMECLUSTER, thereby allowing highly reliable systems to be achieved by using failover.

- Seamless migration from Oracle databases
  Fujitsu Enterprise Postgres provides a compatibility feature with Oracle databases that localizes the correction of existing applications and allows easy migration to Fujitsu Enterprise Postgres.

- Storage Data Protection using Transparent Data Encryption
  Information can be protected from data theft by encrypting data to be stored in the database.

- Data masking for improved security
  The data masking feature changes the returned data for queries from applications, to prevent exposing actual data. This improves security for handling confidential data such as personal information.

- Audit logs for improved security
  Audit logs can be used to counter security threats such as unauthorized access and misuse of privileges for the database.

- Management of access control by confidentiality management
  Confidentiality management feature supports the design/management of access privileges and optimizes access control, thereby enhancing security.

- Enhanced query plan stability
  The following features can control SQL statement query plans:

  - Optimizer hints
  - Locked statistics

  These features are used for curbing performance deterioration caused by changes in SQL statement query plans, such as with mission-critical jobs that emphasize performance stability over improved SQL statement processing performance.

- Increased aggregation performance using the in-memory feature
  The following features help speed up scans even when aggregating many rows.

  - Vertical Clustered Index (VCI)
  - In-memory data

- High-speed data load
  Data from files can be loaded at high speed into Fujitsu Enterprise Postgres tables using the high-speed data load feature.

- High availability by using Connection Manager
  With the Connection Manager features, replication operation can be continued without being aware of the connection destination of the applications.

- Memory usage reduction using Global Meta Cache
  The Global Meta Cache feature loads some of meta cahe information in shared memory. This reduces overall system memory usage.

- Memory usage reduction using Local Meta Cache Limit
  Reduce memory consumption by limiting the metacache that is kept in local memory.

1.1 Flexible Database Recovery

Threats such as data corruption due to disk failure and incorrect operations are unavoidable in systems that use databases. The ability to reliably recover corrupted databases without extensive damage to users when such problems occur is an essential requirement in database systems.

Fujitsu Enterprise Postgres provides the following recovery features that flexibly respond to this requirement:

- Media recovery, which recovers up to the most recent point in time
- Point-in-time recovery, which can recover up to a specific point in time
- Backup/recovery that can integrate with various copy technologies

Media recovery, which recovers up to the most recent point in time

When a disk failure occurs, media recovery can recover data to how it was immediately before the failure.

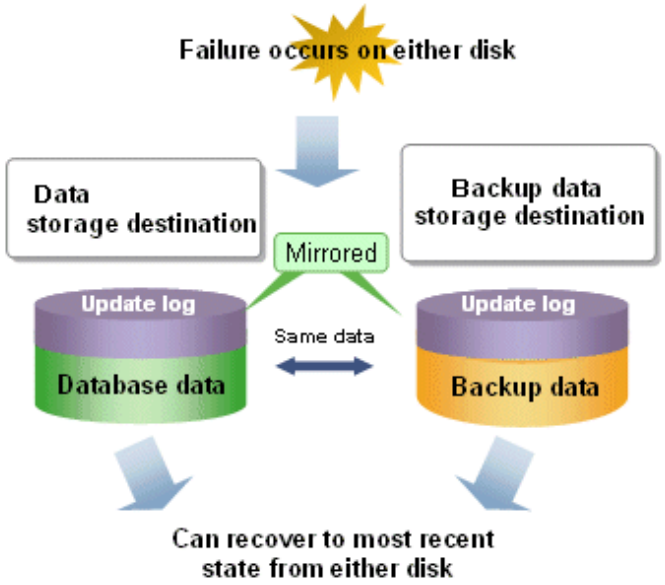
In order to recover the database, Fujitsu Enterprise Postgres accumulates a history of database update operations, such as data additions and deletions, as an update log.

Fujitsu Enterprise Postgres retains a duplicate (mirror image) of the update log after backup execution on the data storage destination and on the backup data storage destination. Therefore, the data on one disk can be used to recover to the most recent state of the database even if a disk failure has occurred on the other.

Media recovery is executed using either a GUI tool provided with Fujitsu Enterprise Postgres (WebAdmin) or server commands.

Information

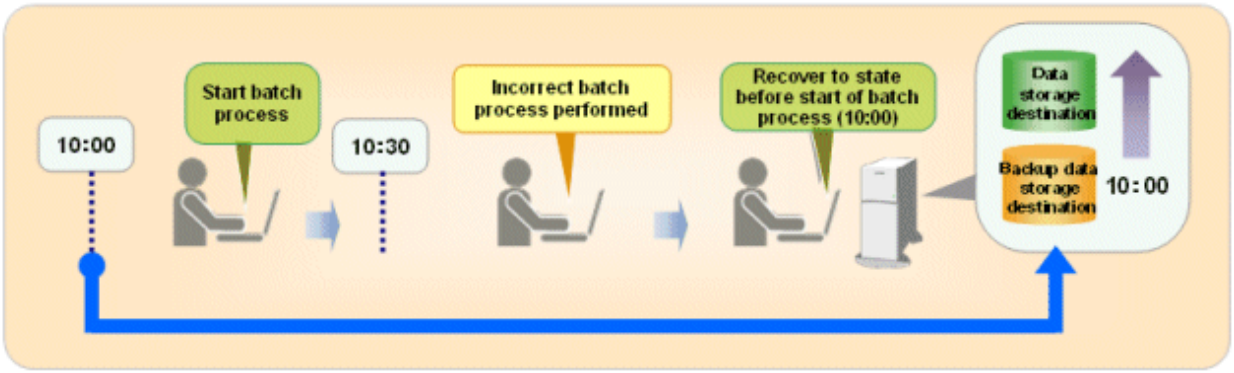
Recovery using WebAdmin requires less time and effort, since WebAdmin automatically determines the scope of the operation.



Point-in-time recovery, which can recover up to a specific point in time

Point-in-time recovery can be used to recover a database that has been updated by an incorrect operation, for example, by specifying any date and time before the incorrect operation.

Point-in-time recovery is executed using Fujitsu Enterprise Postgres server commands.



Backup/recovery that can integrate with various copy technologies

It is possible to back up to the backup data storage destination, or to any backup destination using any copy technology implemented by user exits.

L

For example, by using the high-speed copy feature of the storage device, the processing time for backup of large databases can be greatly reduced.

See

Refer to "Backup/Recovery Using the Copy Command" in the Operation Guide for information on backup/recovery using user exits.

1.2 Simple GUI-Based Installation and Operation Management

Fujitsu Enterprise Postgres provides WebAdmin, which is a GUI tool for a range of tasks, from database installation to operation management. This allows the databases to be used simply and intuitively.

WebAdmin can be used for Fujitsu Enterprise Postgres setup, creating and monitoring a streaming replication cluster, database backups, and for recovery. Depending on the configuration, WebAdmin can be used to manage Fujitsu Enterprise Postgres instances in a single server, or instances spread across multiple servers.

- Setup

  To perform setup using WebAdmin, you must create an instance. An instance is a set of server processes that manage a database cluster (database storage area on the data storage destination disk). Instances can be created easily and with only minimal required input, because the tool automatically determines the optimal settings for operation.

- Database backup/recovery

  Database backup and recovery can be performed using simple GUI operations.

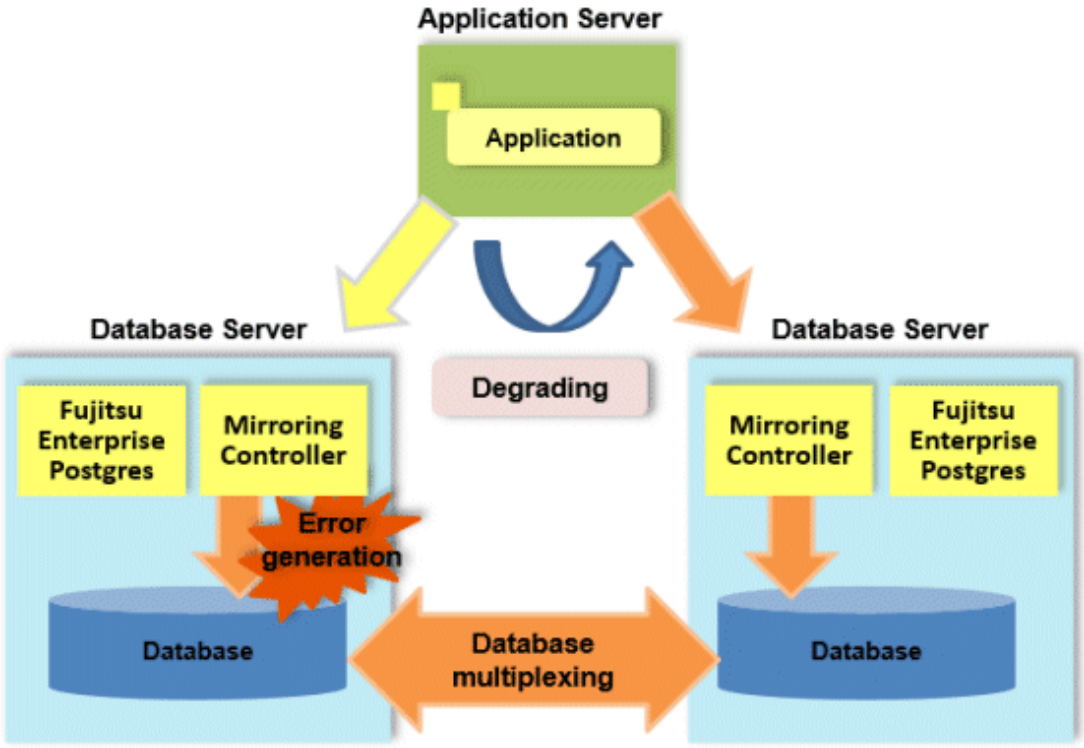
  In particular, Fujitsu Enterprise Postgres can automatically identify and isolate the location of errors. This simplifies the recovery process and enables faster recovery.

In addition, Fujitsu Enterprise Postgres provides the following expanded features in pgAdmin:

- NCHAR type
- Expanded trigger definition
  - REPLACE feature
  - Function call feature

1.3 High Reliability with Database Multiplexing

It is vital for systems that use databases to protect data from damage or loss caused by a range of factors such as hardware and software errors. Database multiplexing protects important data and enables highly reliable database operation.



Fujitsu Enterprise Postgres not only mirrors a database using the PostgreSQL streaming replication feature, but also provides simplified switchover and standby disconnection features as well as a feature to detect faults in elements that are essential for the continuity of database process, disk, network, and other database operations.

Even if a switchover is performed, the client automatically distinguishes between the primary and standby servers, so applications can be connected transparently regardless of the physical server.

The Mirroring Controller option enables the primary server (the database server used for the main jobs) to be switched automatically to the standby server if an error occurs in the former.

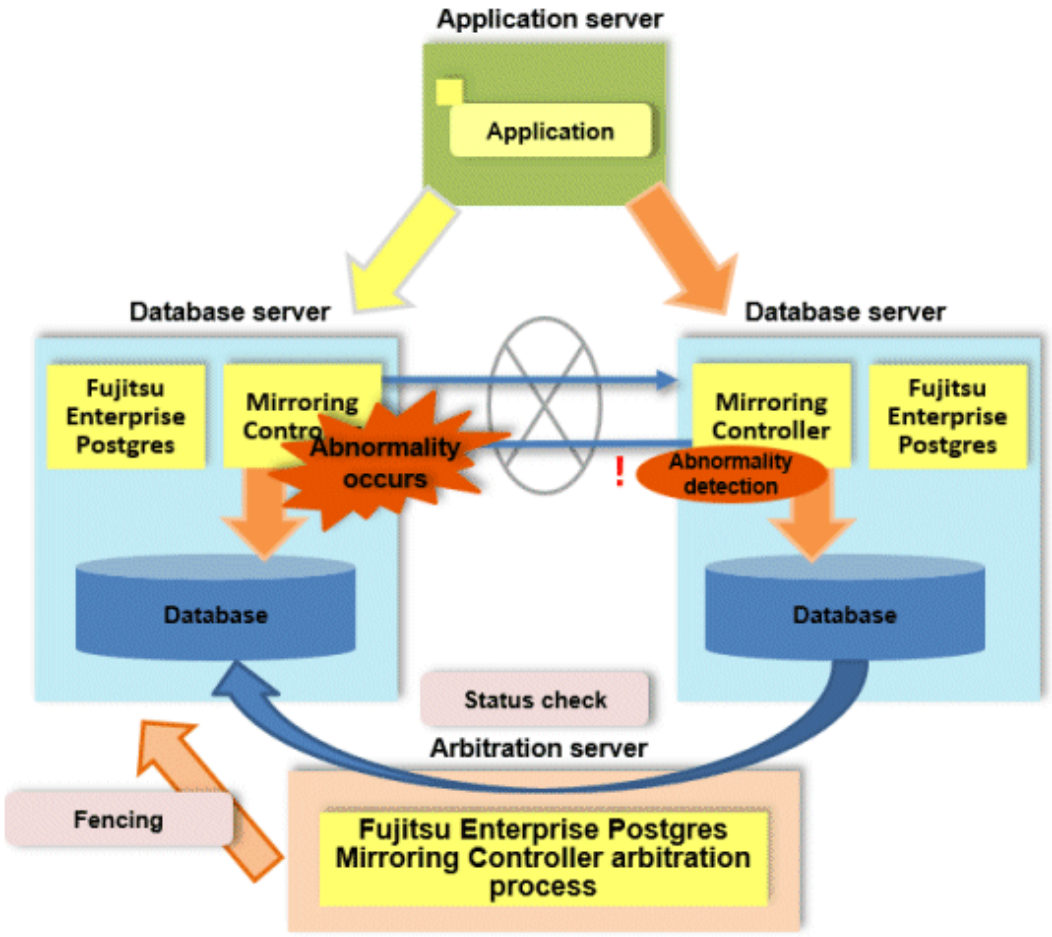
In addition, by using the data on the standby server, reference jobs such as data analysis and form output can be performed in parallel to the jobs on the primary server.

Operation using the arbitration server

Mirroring Controller may not be able to correctly determine the status of the other server if there is a network issue between database servers or a server is in an unstable state. As a result, both servers will temporarily operate as primary servers, so it may be possible to perform updates from either server.

The Server Assistant is a feature that objectively checks the status of database servers as a third party and isolates (fences) unstable servers in such cases.

In database multiplexing mode, the Server Assistant is made available by adding a new server (arbitration server) on which the Server Assistant is installed. Using an arbitration server can prevent the issue mentioned above (both servers temporarily operating as primary servers) and enables highly reliable operation.



See

Refer to "Database Multiplexing Mode" in the Cluster Operation Guide (Database Multiplexing) for information on the database multiplexing.

L

1.4 High Reliability Using Failover Integrated with the Cluster Software

If the system stops, services are interrupted until recovery is complete. In large-scale systems, the interruption takes longer, and may cause significant disruption for many people receiving the services.

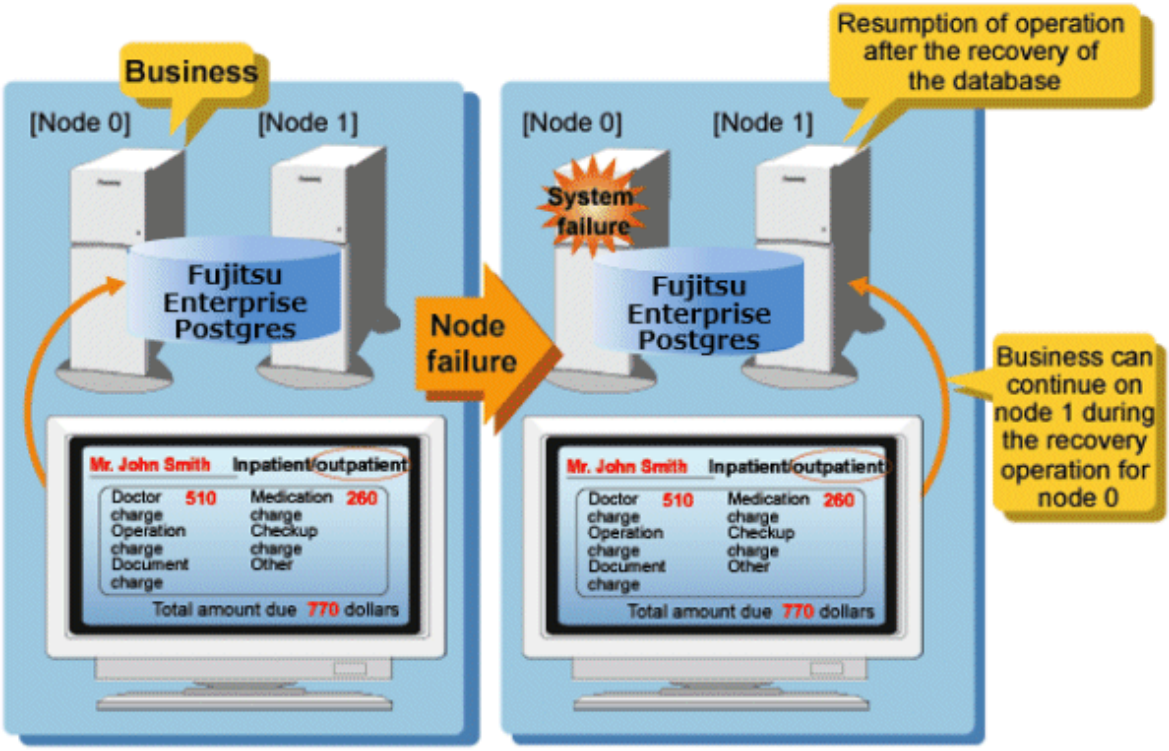
In Fujitsu Enterprise Postgres, the failover feature integrated with the cluster software can minimize the system stoppage time when an issue occurs.

Medical accounting system

Some hospitals with a large number of patients manage and operate the various data required for accounting in a database. If this accounting system stops and takes several minutes to recover, it is expected that this will have a significant impact.

But if failover is applied to this kind of system and an issue occurs on the operating server, it is quickly switched and the standby server takes over operation, so that services are provided without interruption.

The example below illustrates a medical accounting system using failover.



See

Refer to the Cluster Operation Guide (PRIMECLUSTER) for information on the failover feature integrated with the cluster software.

1.5 Seamless Migration from Oracle Databases

Fujitsu Enterprise Postgres supports Orafce, to provide compatibility with Oracle databases.

Using the compatibility feature reduces the cost of correcting existing applications and results in easy database migration.

See

Refer to "Compatibility with Oracle Databases" in the Application Development Guide for information on compatible features.

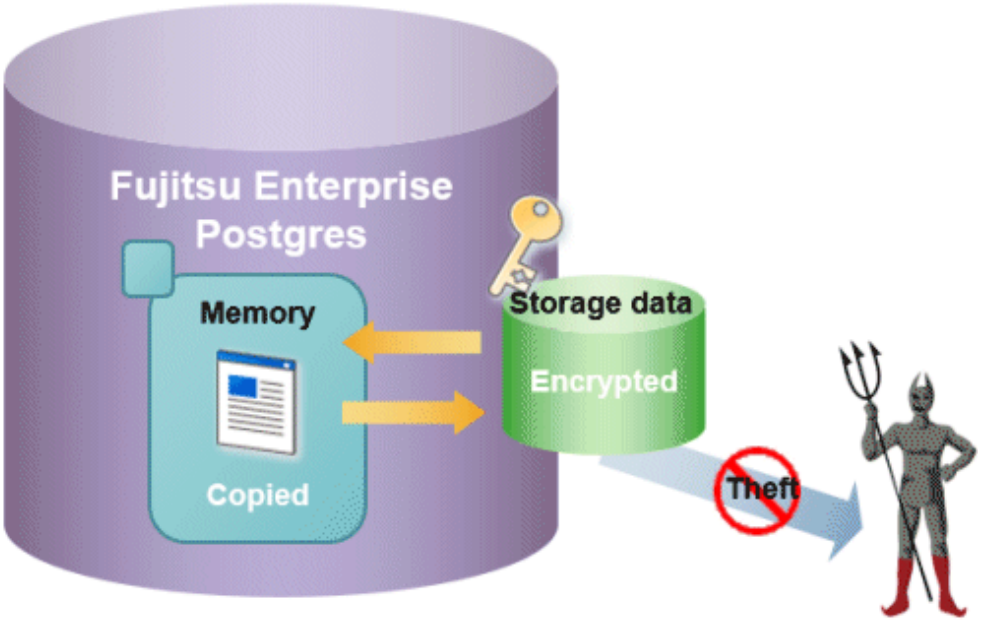
1.6 Storage Data Protection Using Transparent Data Encryption

The encryption of data to be stored in a database is essential under the following encryption requirements of PCI DSS (Payment Card Industry Data Security Standard), the data security standard of the credit industry:

- Confidential information (such as credit card numbers) can be encrypted.
- The encryption key and data are managed as separate entities.
- The encryption key is replaced at regular intervals.

To satisfy these requirements, Fujitsu Enterprise Postgres provides a transparent data encryption feature. Note that PostgreSQL uses an encryption feature called pgcrypto, which can also be used in Fujitsu Enterprise Postgres, but requires

applications to be modified. Therefore, we recommend using Fujitsu Enterprise Postgres's transparent data encryption feature.



The transparent data encryption feature also allows you to choose an external key management system as the storage location for your encryption keys.

See

Refer to "Protecting Storage Data Using Transparent Data Encryption" or "Using Transparent Data Encryption with Key Management Systems as Keystores" in the Operation Guide for information about transparent data encryption.

Refer to "Protecting Storage Data Using Transparent Data Encryption" in the Operation Guide for information about transparent data encryption.

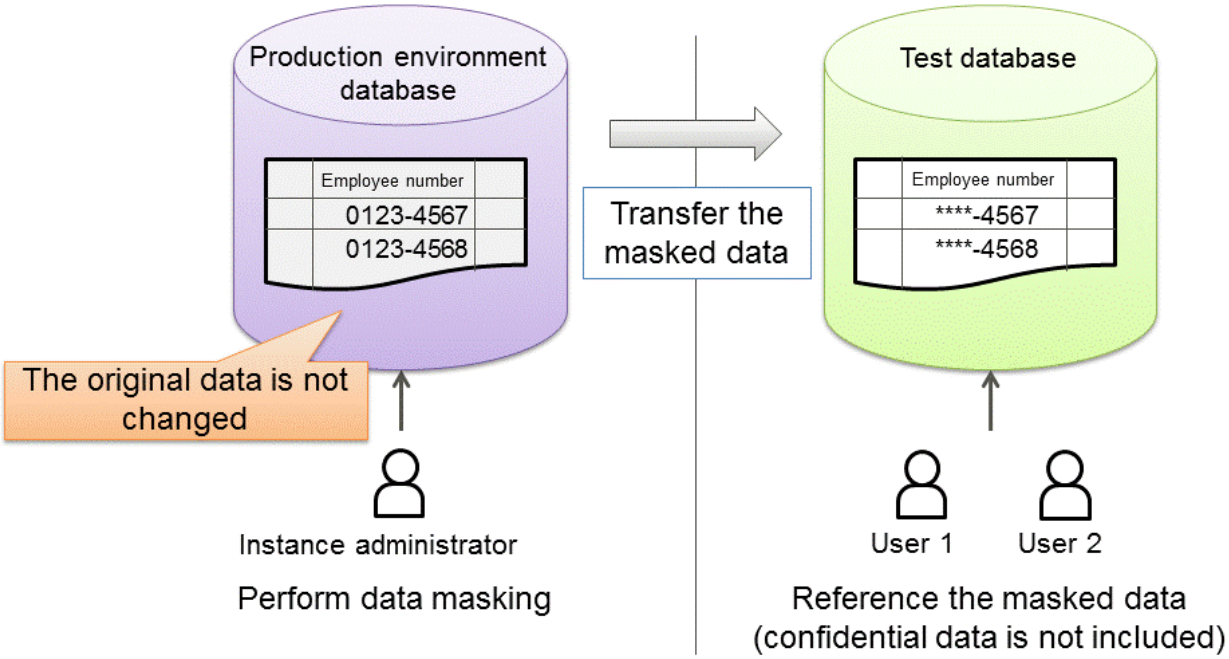
1.7 Data Masking for Improved Security

Fujitsu Enterprise Postgres provides a data masking feature that protects data to maintain security of data handled in systems.

The data masking feature changes the returned data for queries from applications and makes it available for reference without exposing the actual data.

For example, for a query of employee data, digits except the last four digits of an eight-digit employee number can be changed to "*" so that it can be used for reference.

Also, the data changed by the data masking feature can be transferred to a test database so that users who perform testing or development can reference the data. As production data should not be used in a test or development environment because of the risk of data leakage, this feature enables data that is similar to actual production data to be safely used in those environments.



See

Refer to "Data Masking" in the Operation Guide for information on data masking.

1.8 Security Enhancement Using Audit Logs

Details relating to database access can be retrieved in audit logs. The audit log feature can be used to counter security threats such as unauthorized access to the database and misuse of privileges.

In PostgreSQL, logs output as server logs can be used as audit logs by using the log output feature. There are, however, logs that cannot be analyzed properly, such as SQL runtime logs, which do not output the schema name. Additionally, because the output conditions cannot be specified in detail, log volumes can be large, which may lead to deterioration in performance.

The audit log feature of Fujitsu Enterprise Postgres enables retrieval of details relating to database access as an audit log by extending the feature to pgaudit. Additionally, audit logs can be output to a dedicated log file or server log. This enables efficient and accurate log monitoring.

Note

This feature can only be used in Advanced Edition.

See

Refer to "Audit Log Feature" in the Security Operation Guide for details.

1.9 Management of Access Control by Confidentiality Management

The confidentiality management feature supports the realization of access control according to the confidentiality level of data. We also support the work of confirming that operations are being performed according to access control.

Note

This feature can only be used in Advanced Edition.

Access control

Access to sensitive data must be restricted in order to comply with the laws and rules governing data protection regulations. However, designing the access control is not easy in databases with diverse data and users performing diverse tasks. This is because for every combination of all database object and all role that can access the data, you must decide whether to allow access and define it in the database.

In our real world, we don't do that. For example, in a business group data with the same confidentiality level, and group several roles accessing to that data. After that, it makes more sense to consider whether access should be granted for combinations of group of data and group of roles. Because once data is added, deciding which group it belongs to naturally determines who can access it. When adding a role, it is enough to think about which role's group (called a confidentiality group) to add it to. The confidentiality management feature supports such a natural design.

Also, data belonging to a high level of confidentiality may need to be protected against unauthorized access to physical media and files, as well as access from users who can log into the database. The confidentiality management feature can force the encryption of tables belonging to high confidentiality levels. Similarly, you can force roles that belong to a confidentiality group to have attributes less than or equal to those set for the confidentiality group. Such table encryption and role management can also be designed naturally with this feature.

Inspection of operation

In order to comply with the laws and rules that define data protection regulations, it is necessary to ensure that the database is operated safely as designed. If you are using the confidentiality management feature, you do not have to worry about such things. However, if table or role definitions are changed without using this feature, it must be detected timely. The confidentiality management feature does not prohibit or detect such acts. Instead, use audit logs to detect such changes, etc.

However, even if they detect it, they may forget to deal with it. In order to prevent this, it is necessary to periodically check the differences between the confidentiality levels and confidentiality groups and the actual table and role definitions. At that time, you can use theconfidentiality management feature provided to obtain the difference.

See

Refer to "Confidentiality Management" in the Security Operation Guide for details.

1.10 Enhanced Query Plan Stability

Fujitsu Enterprise Postgres estimates the cost of query plans based on SQL statements and database statistical information, and selects the least expensive query plan. However, like other databases, Fujitsu Enterprise Postgres does not necessarily select the most suitable query plan. For example, it may suddenly select unsuitable query plan due to changes in the data conditions.

In mission-critical systems, stable performance is more important than improved performance, and changes in query plans case to be avoided. In this situation, the following features can stabilize query plans:

- Optimizer hints

  You can use pg_hint_plan to specify a query plan in each individual SQL statement.

- Locked statistics

  You can use pg_dbms_stats to lock statistical information per object, such as a database, schema, or table.

See

Refer to "Optimizer Hints" in the Application Development Guide for information on optimizer hints.

Refer to "Locked Statistics" in the Application Development Guide or information on locked statistics.

Note

Use the features provided when Fujitsu Enterprise Postgres is installed for optimizer hints and locked statistical information. Fujitsu Enterprise Postgres does not support other similar open-source features.

1.11 Increased Aggregation Performance Using the In-memory Feature

Fujitsu Enterprise Postgres provides the in-memory feature, which uses columnar index and memory-resident data. This reduces disk I/Os and enhances aggregation performance.

Note

This feature can only be used in Advanced Edition.

Columnar index

Many aggregation processes may require a large portion of data in a particular column. However, traditional row data structure reads unnecessary columns, resulting in inefficient use of memory and CPU cache, and slower processing. Fujitsu Enterprise Postgres provides a type of columnar index, VCI (Vertical Clustered Index). This addresses the above issues, and enhances aggregation performance.

VCI provides the following benefits:

- Minimizes impact on existing jobs, and can perform aggregation using job data in real time.
- Provided as an index, so no application modification is required.
- Stores data also on the disk, so aggregation jobs can be quickly resumed using a VCI even if a failure occurs (when an instance is restarted).
- If the amount of memory used by VCI exceeds the set value, aggregation can still continue by using VCI data on the disk.

It also provides the features below:

- Disk compression
  Compresses VCI data on the disk, minimizing required disk space. Even if disk access is required, read overhead is low.
- Parallel scan
  Enhances aggregation performance by distributing aggregation processes to multiple CPU cores and then processing them in parallel.

In-memory data

The following features keep VCI data in memory and minimize disk I/Os on each aggregation process.

- Preload feature
  Ensures stable response times by loading VCI data to memory before an application scans it after the instance is restarted.
- Stable buffer feature
  Reduces disk I/Os by suppressing VCI data eviction from memory by other job data.

Purposes of this feature

This feature has a data structure that can efficiently use the newly added resources, and aims to enhance the existing aggregation processing in normal operations to be faster than parallel scan. It shares the same purpose of enhancing

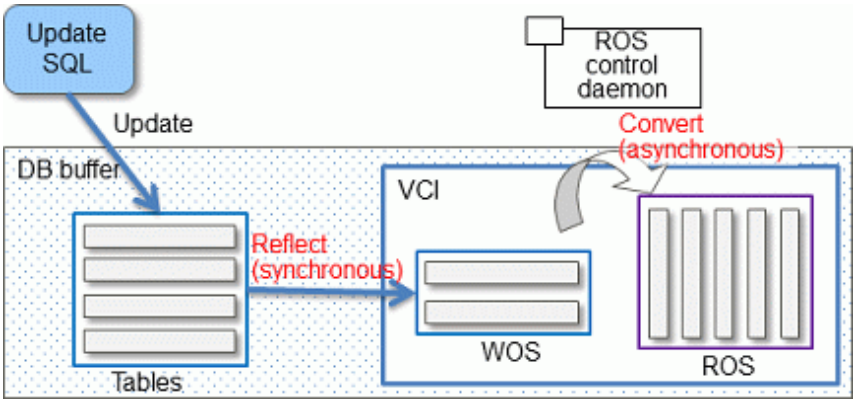
aggregation performance with the parallel scan feature that is provided separately, but differs in that it speeds up nightly batch processes by utilizing available resources.

VCI architecture

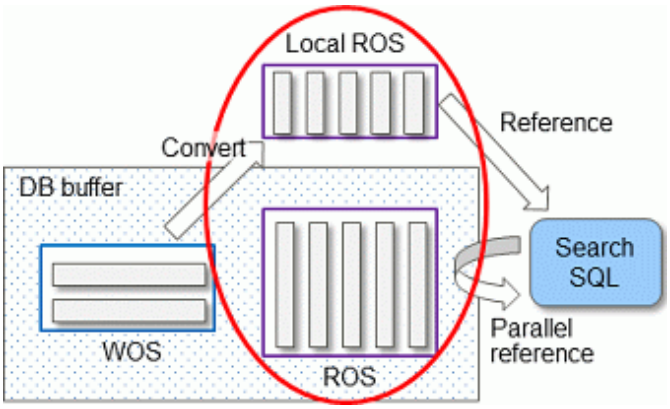
This section briefly explains VCI architecture as it contains basic terminology required, for example, when setting parameters.

Update and aggregation operations to enable real time use of job data are described.

VCI has write buffer row-based WOS (Write Optimized Store) in addition to the columnar data structure ROS (Read Optimized Store). Converting each update into a columnar index has a significant impact on the update process response times. Therefore, data is synchronously reflected to the row-based WOS when updating. After a certain amount of data is stored in WOS, the ROS control daemon asynchronously converts it to ROS. As above, the entire VCI is synchronized with the target table column, minimizing update overhead.



The same scan results can be obtained without a VCI by using WOS in conjunction with ROS. More specifically, WOS is converted to Local ROS in local memory for each aggregation process, and aggregated with ROS.



See

Refer to "Installing and Operating the In-memory Feature" in the Operation Guide for information on installation and operation of VCI.

Refer to "Scan Using a Vertical Clustered Index (VCI)" in the Application Development Guide for information on scan using a VCI.

1.12 High-Speed Data Load

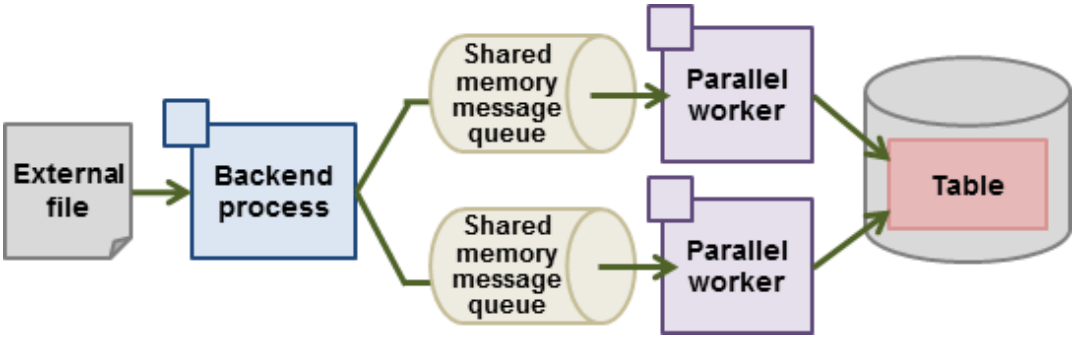
High-speed data load executes COPY FROM commands using multiple parallel workers. Because conversion of data from the external file to the appropriate internal format, table creation, and index creation are performed in parallel, it is possible to load large volumes of data at high speed.

Note

This feature is available only in the Advanced Edition.

Architecture of high-speed data load

High-speed data load is required for parameter setting and resource estimation, so a brief description of its architecture is provided below.



High-speed data load uses a single backend process collaborating with multiple parallel workers to perform data load in parallel. Data is exchanged between the backend process and parallel workers via shared memory message queues. The backend process distributes the loaded data of external files to multiple parallel workers. Each parallel worker then converts the data loaded from the shared memory message queue into the appropriate internal format, and inserts it into the table. If the table has indexes, their keys are extracted and inserted into the index page.

See

Refer to "High-Speed Data Load" in the Operation Guide for details.

1.13 High availability by using Connection Manager

The Connection Manager provides the following features. You can use these features to increase system availability.

Note

This feature can only be used in Advanced Edition.

Heartbeat monitoring feature

Detects kernel panics between the server running the client and the server running the instance, physical server failures, and inter-server network link downs, and notifies the client or instance. The client is notified as an error event through the SQL connection, and the instance will be notified in the form of a force collection of SQL connections with clients that are out of service.

Transparent connection support feature

When an application wants to connect to an instance of an attribute (Primary/Standby) configured with replication, it can do so without knowing which server the instance is running on.

Information

The available client drivers for Connection Manager are libpq (C language library), ECPG (embedded SQL in C), ECOBPG (embedded SQL in COBOL), ODBC driver and JDBC driver.

See

Refer to the Connection Manager User's Guide for details.

1.14 Memory Usage Reduce with Meta cache Reduction and Limit

When executing SQL, you must refer to the definition of the table or index you want to access. These definitions are cached in the local memory of the backend process separately from the shared memory buffer of the database on shared_buffers because they are referenced each time SQL is executed. The direct definition is a tuple of system catalogs. The cache for this tuple is called "Catalog Cache". A structure that makes the definition easy to use is called a "Relations Cache". And in Fujitsu Enterprise Postgres, these two are collectively called "Meta Cache".

The meta cache will be kept indefinitely for performance reasons. In a large database, a single backend process accesses a large number of tables and so on, which results in a large meta-cache for the backend process. As a result, the sum of the local memory of the backend process may exceed the realistic memory size.

On the other hand, the feature to reduce the meta cache for the entire instance by sharing the meta cache between backend processes is the Global Meta Cache feature. The current Global Meta Cache feature only shares the catalog cache. Therefore, the metacache in Global Meta Cache now refers to the catalog cache.

What you still cannot share using the current Global Meta Cache feature and need to keep in local memory is the information (Meta cache header) and relation cache to access Global Meta Cache in shared memory. If you do not use the Global Meta Cache feature, keep the catalog and relation caches in local memory. The meta cache held in local memory is called the "Local Meta Cache". The feature to limit the size of a Local Meta Cache by removing it if it has not been accessed for a long time is the Local Meta Cache limit feature.

The Global Meta Cache feature and the Local Meta Cache limit feature can be used together to provide the strictest control over memory consumption. Of course, you can use only one or the other.

However, the Global Meta Cache feature has several percent overhead to access shared memory. The Local Meta Cache limit feature also causes the overhead of reholding the metacache because it may discard the previously held metacache. Therefore, consider using these features when your estimates do not allow for memory consumption.

1.14.1 Memory Usage Reduction Using Global Meta Cache

The Global Meta Cache feature cache the meta cache in shared memory. The meta cache on shared memory is called the Global Meta Cache (GMC).

Without this feature, the meta cache was cached in per-process memory. Therefore, there was a problem increase in memory usage in environments with large databases and large numbers of connections. The Global Meta Cache feature enables sharing of meta caches on shared memory, thereby reducing overall system memory usage.

Note

This feature is available only in the Advanced Edition.

Meta cache

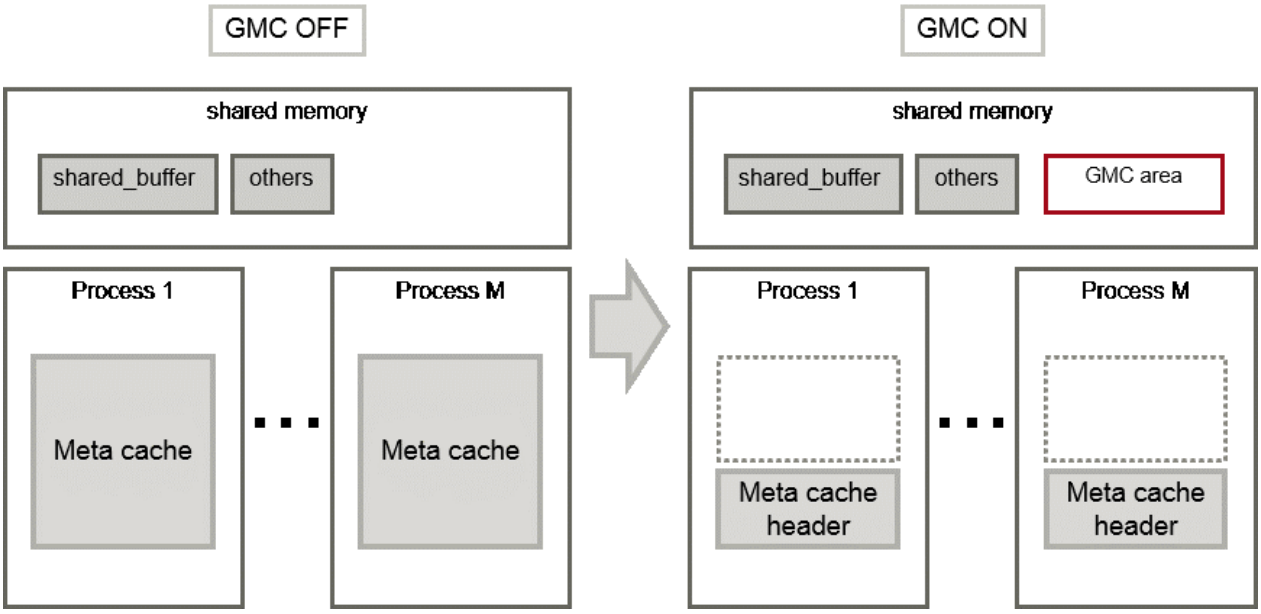
Processing a query involves parsing the query, creating the plan, executing the plan, and so on. PostgreSQL process accesses the system catalog to perform these steps. Once accessed, the system catalog tuples are cached in per-process memory. The

direct definition is one tuple of system catalogs. Each process performs faster query processing by searching the meta cache instead of searching for the required tuples in the system catalog each time.

The meta cache usage increases in proportion to the number of tables and columns accessed. It is cached on a per-process basis, so the system's overall meta cache usage increases in proportion to the number of connections.

Architecture of Global Meta Cache feature

Describes the architecture of the Global Meta Cache feature.



When the GMC feature is on, the per-process meta cache is cached in the GMC area on shared memory. Reference to the GMC area and process-specific work information is cached in the memory of each process. PostgreSQL process searches the meta cache for each process and accesses the GMC based on the reference information. If there is no reference information in the process's memory, it searches the GMC area. If the GMC area also does not have a corresponding meta cache, it accesses the system catalog to create meta cache.

Also, sharing the meta cache does not cause any loss of data consistency. If the system catalog or table definition changes while a transaction is running, the cache deletion or creation does not affect outside of the process running the transaction. After the transaction commits, the GMC area cache is deleted or created. If other transactions are referencing the cache when GMC is tried to be deleted, the deletion is deferred until there are no more references. After a commit, a new transaction sees the new cache instead of the old one.

See

Global Meta Cache feature is disabled by default. Refer to "Global Meta Cache" in the Operation Guide for information how to decide whether introduce it or not and usage.

1.14.2 Memory Usage Reduction Using Local Meta Cache Limit

Local Meta Cache Limit feature limits the size of a Local Meta Cache by removing it if it has not been accessed for a long time.

Of the definitions that SQL accesses, the main factors that make the Local Meta Cache bloat are tables and indexes. In addition, table column definitions are also maintained as a catalog cache.

For example, in a system where one long-lived connection is shared by various businesses, one connection (that is, backend process) will access many tables. If there are 3,000 such connections, and each connection accesses a table of 50,000, the total amount of memory consumed by the 3,000 backend processes may be a few terabytes.

In such a case, using this feature may reduce it to about several tens of gigabytes.

Note

This feature is available only in the Advanced Edition.

Architecture of Local Meta Cache Limit feature

When this feature is enabled, the caching strategy is to keep the cache as long as possible within the specified upper limit. If holding a new cache exceeds the limit, consider locality of reference and delete the cache from the one with the longest unreferenced time.

However, because the cache used by active transactions cannot be deleted, if a transaction uses a large number of caches, the cache may be held above the limit. In this case, delete the cache at the end of the transaction.

See

Local Meta Cache limit feature is disabled by default. Refer to "Local Meta Cache Limit" in the Operation Guide for information how to decide whether introduce it or not and usage.

Appendix A List of Features

The following table lists the main features provided by Fujitsu Enterprise Postgres.

Category	Feature	Linux		Windows	
		AE	SE	AE	SE
Fujitsu-developed software technology	WebAdmin (Rapid setup, One-click recovery)	Y	Y	Y	Y
Improved reliability and availability	Database multiplexing	Y	N	Y	N
	Failover (integration with PRIMECLUSTER)	Y (*1)	Y (*1)	N	N
	Backup/recovery using user exits	Y	N	Y	N
	Connection Manager	Y	N	Y	N
Application development	Embedded SQL integration Java integration ODBC integration .NET Framework integration	Y	Y	Y	Y
	Features compatible with Oracle databases	Y	Y	Y	Y
Security	Transparent data encryption	Y	Y	Y	Y
	Data masking	Y	Y	Y	Y
	Audit log	Y	N	Y	N
	Confidentiality management	Y	N	Y	N
Performance	In-memory feature	Y	N	Y	N
	High-speed data load	Y	N	Y	N
	Global Meta Cache	Y	N	Y	N
	Local Meta Cache Limit	Y	N	Y	N
Monitoring	Database Monitoring with Datasentinel (*2)	Y	Y	Y	Y
Performance tuning	Optimizer hints	Y	Y	Y	Y
	Fixed statistical information	Y	Y	Y	Y

Y: Provided

N: Not provided

*1: Supported on RHEL only.

*2: Monitoring is possible in conjunction with Datasentinel. For more information, see the Datasentinel documentation available at:

https://docs.datasentinel.io/manual

Appendix B OSS Supported by Fujitsu Enterprise Postgres

The OSS supported by Fujitsu Enterprise Postgres is listed below.

OSS name	Version and level	Platform		Description	Reference
		Linux	Windows		
PostgreSQL	15.0	Y	Y	Database management system	PostgreSQL Documentation
orafce	3.25.1	Y	Y	Oracle-compatible SQL features	"Compatibility with Oracle Databases" in the Application Development Guide
Pgpool-II	4.4.0	Y	N	Failover, connection pooling, load balancing, etc.	"Pgpool-II" in the Installation and Setup Guide for Server
oracle_fdw	2.5.0	Y	Y	Connection to the Oracle database server	"oracle_fdw" in the Installation and Setup Guide for Server
pg_statsinfo	-	Y	N	Collection and accumulation of statistics	"pg_statsinfo" in the Installation and Setup Guide for Server
pg_hint_plan	14.1.4.0	Y	Y	Tuning (statistics management, query tuning)	- "pg_hint_plan" in the Installation and Setup Guide for Server - "Optimizer Hints" in the Application Development Guide
pg_dbms_stats	-	Y	Y		- "pg_dbms_stats" in the Installation and Setup Guide for Server - "Locked Statistics" in the Application Development Guide
pg_repack	1.4.8	Y	N	Table reorganization	"pg_repack" in the Installation and Setup Guide for Server
pg_rman	1.3.14	Y	N	Backup and restore management	"pg_rman" in the Installation and Setup Guide for Server
pgAdmin4	6.15	N	Y	Operation and development GUI	Operation Guide
pgBadger	12.0	Y	N	Log analysis	"pgBadger" in the Installation and Setup Guide for Server
pg_bigm	1.2	Y	N	Full-text search (multibyte)	"pg_bigm" in the Installation and Setup Guide for Server
PostgreSQL JDBC driver	42.5.0	Y	Y	JDBC driver	"JDBC Driver" in the Application Development Guide

OSS name	Version and level	Platform		Description	Reference
		Linux	Windows		
psqlODBC	13.02.0000	Y	Y	ODBC driver	"ODBC Driver" in the Application Development Guide
Npgsql	6.0.7	N	Y	.NET data provider	".NET Data Provider" in the Application Development Guide

Y: Supported

N: Not supported

Appendix C Features that can be Used on Servers Other than the Database Server

This chapter explains the configuration and operating environment of features to be installed and used on servers other than the database server when used in conjunction with the Fujitsu Enterprise Postgres database server.

In this chapter, Fujitsu Enterprise Postgres programs are referred to as server programs.

Below are features to be installed and used on servers other than the database server:

- WebAdmin
- Server Assistant
- Pgpool-II (failover, connection pooling, and load balancing)

C.1 WebAdmin

If there is only one database server, WebAdmin is normally installed on the same server as the database (the WebAdmin program can be installed at the same time as the server program).

If there are multiple database servers, database server instances can be managed collectively if a dedicated WebAdmin server is used. In this case, the WebAdmin program is installed on the WebAdmin server, and the server program and WebAdmin program are installed on the database server.

See

- Refer to "1.2 Simple GUI-Based Installation and Operation Management" for information on WebAdmin.
- Refer to "Determining the Preferred WebAdmin Configuration" in the Installation and Setup Guide for Server for information on the server configuration when using WebAdmin.
- Refer to "Required Operating System" in the Installation and Setup Guide for Server for information on the operating environment of WebAdmin.

C.2 Server Assistant

To use the Server Assistant, the Server Assistant program is installed on a dedicated server (arbitration server).

See

- Refer to "Overview of Database Multiplexing Mode" in the Cluster Operation Guide (Database Multiplexing) for information on the Server Assistant and the server configuration.
- Refer to "Required Operating System" in the Installation and Setup Guide for Server Assistant for information on the operating environment of the Server Assistant.

C.3 Failover, Connection Pooling, and Load Balancing Features of Pgpool-II

Pgpool-II is software that is placed between the database server and database client to relay the connection.

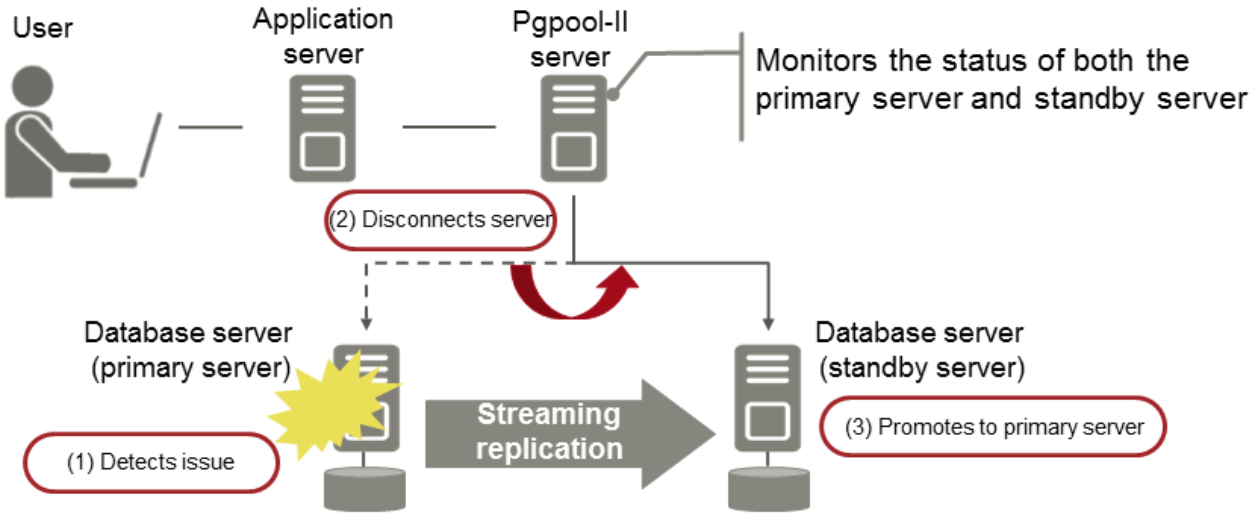
Pgpool-II provides the failover, connection pooling, and load balancing features for use during streaming replication.

Failover

In PostgreSQL, a database can be made redundant (building a high availability system) using synchronous streaming replication.

If the database server of either the primary server or standby server fails or is no longer accessible when using synchronous streaming replication, jobs will stop.

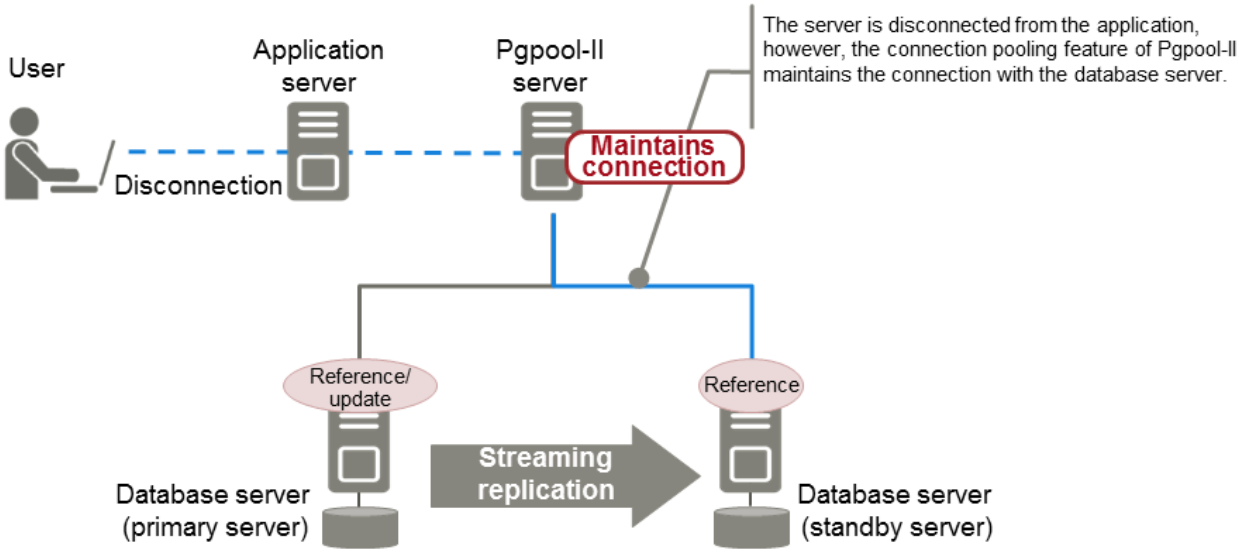
Failover monitors the status of each database and automatically disconnects the server when an error occurs. As a result, jobs can continue uninterrupted on the remaining server.



Connection pooling

This feature maintains (pools) the connection established with the database server, and reuses that connection each time a new connection with the same properties (user name, database, and protocol version) arrives.

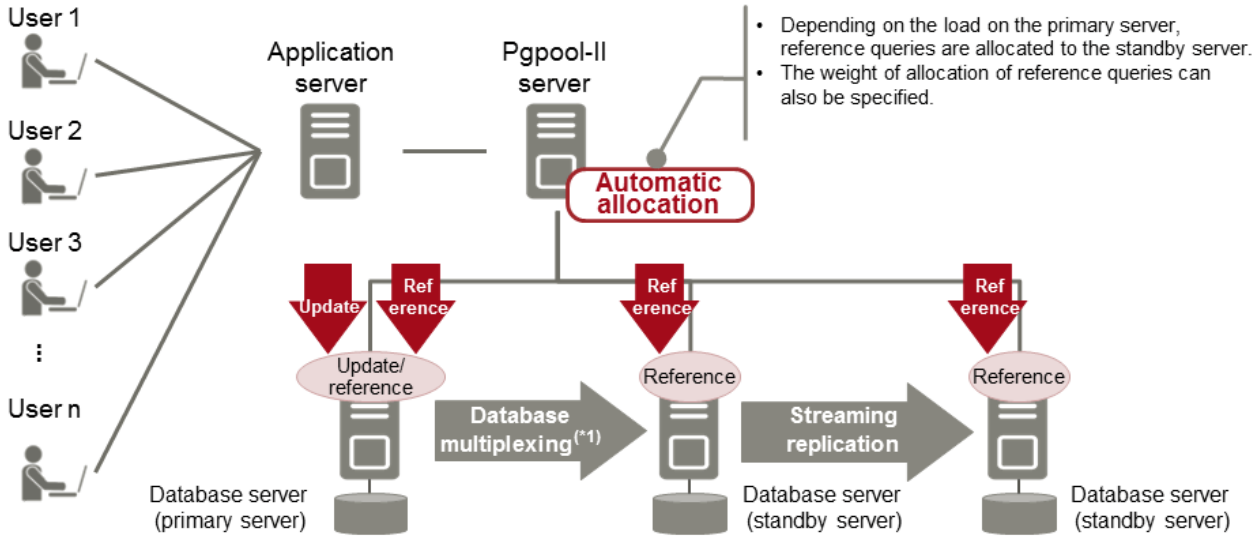
Connection pooling reduces the connection overhead for the database server, improving throughput of the whole system.



Load balancing

This feature distributes reference queries to multiple database servers, improving throughput of the whole system.

By combining load balancing with the Fujitsu Enterprise Postgres database multiplexing feature or the PostgreSQL streaming replication feature, load on the database server is reduced.



*1: The arbitration server used during database multiplexing has been omitted from this document.

See

- Refer to "System configuration when using Pgpool-II" in the Installation and Setup Guide for Server for information on the server configuration when using Pgpool-II.
- Refer to "Required Operating System" in the Installation and Setup Guide for Server for information on the operating environment of Pgpool-II.

Index

[C]
Columnar index..... 11
compatibility with Oracle databases.....7
Connection Manager.....13

[D]
Database Multiplexing.....4
Data Masking for Improved Security..... 8

[F]
Flexible Database Recovery..... 2

[G]
Global Meta Cache..... 14

[H]
High-Speed Data Load..... 12

[I]
In-memory data.....11

[M]
Media recovery..... 2

[O]
Oracle Database.....7

[P]
Point-in-time recovery.....3

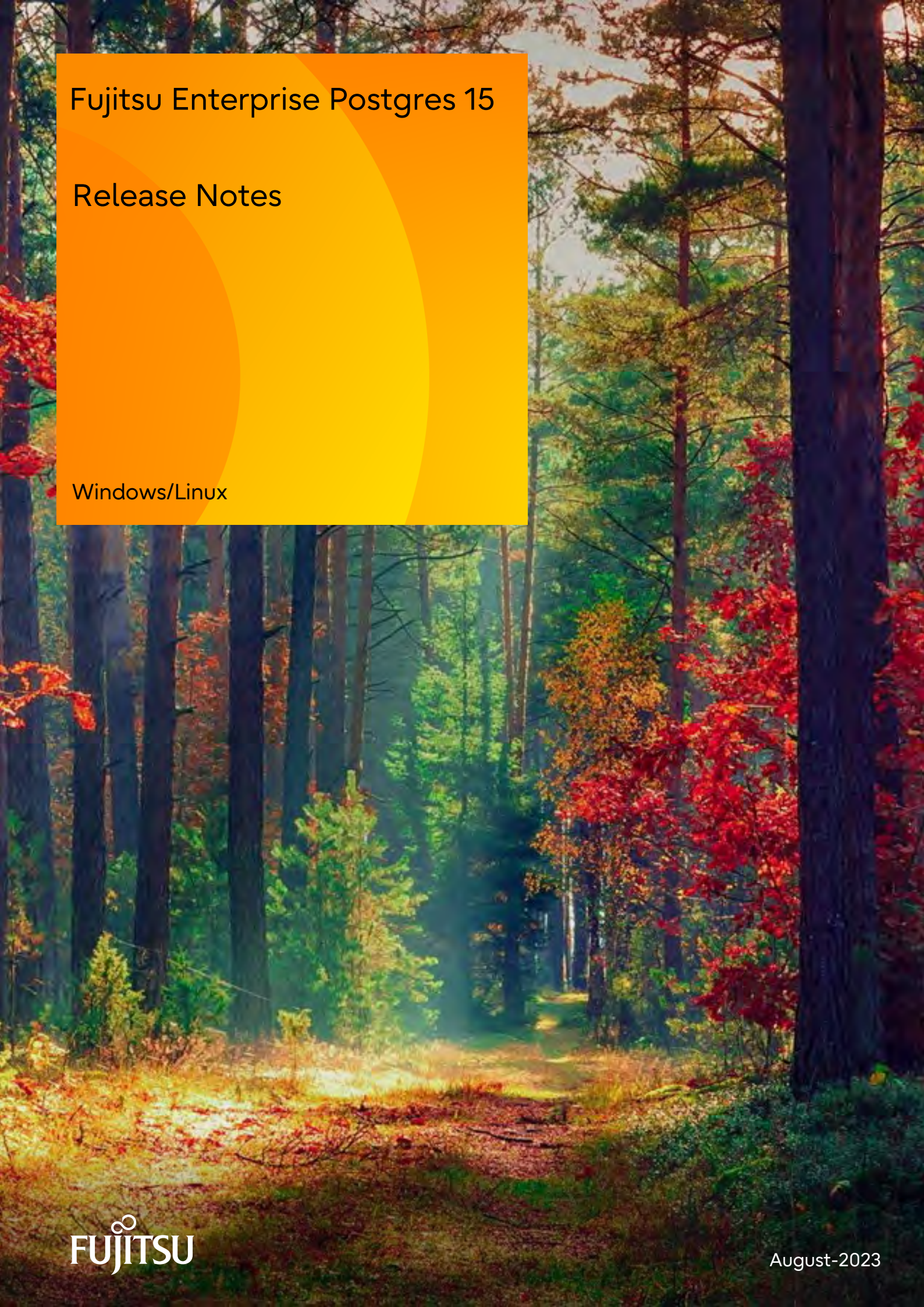
[S]
Security Enhancement Using Audit Logs..... 9

[T]
Transparent Data Encryption.....7

Fujitsu Enterprise Postgres 15

Release Notes

Windows/Linux



FUJITSU

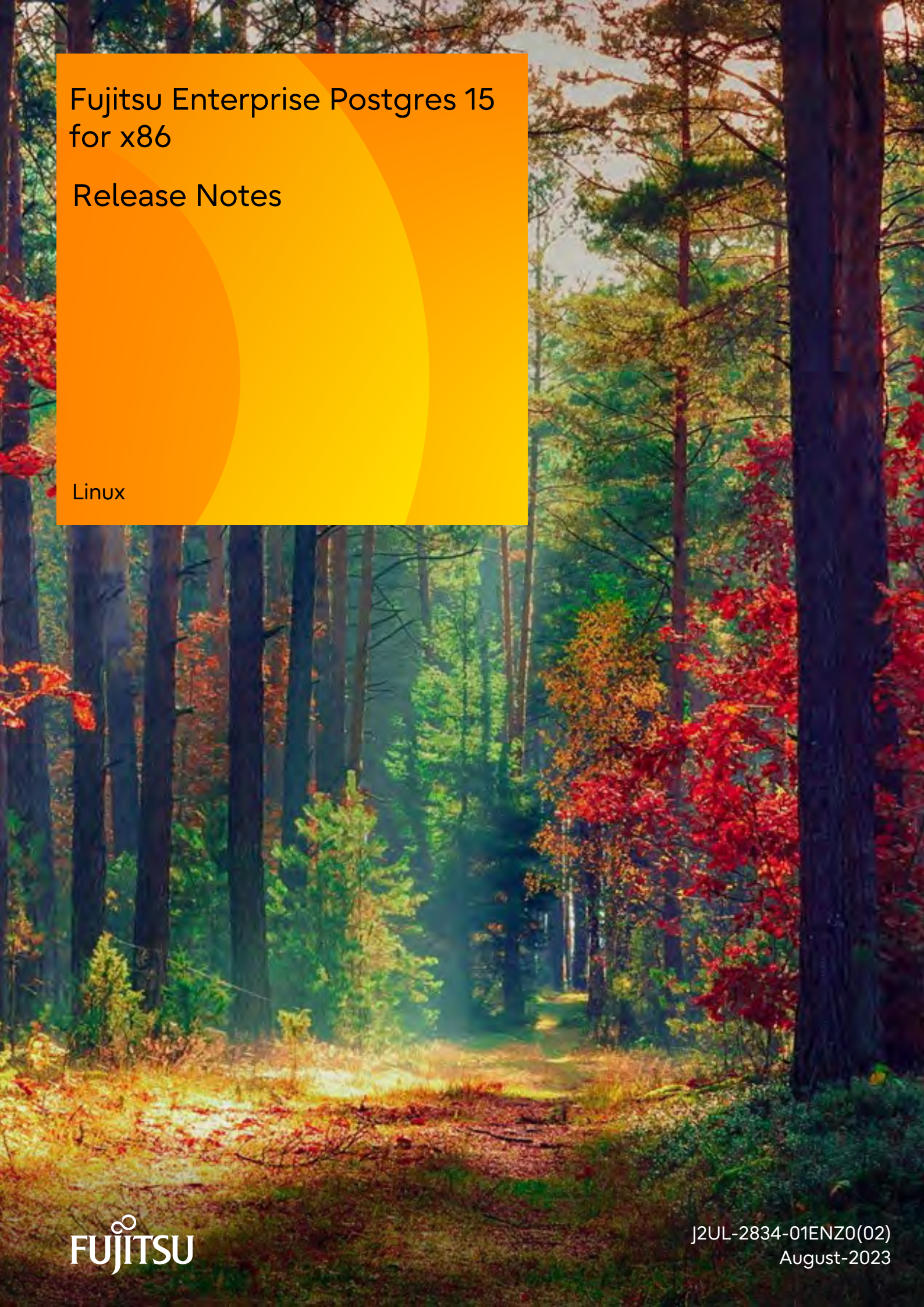
August-2023

Linux >
Windows >

Fujitsu Enterprise Postgres 15 for x86

Release Notes

Linux



FUJITSU

J2UL-2834-01ENZ0(02)
August-2023

Preface

Purpose of this document

This document provides release information for Fujitsu Enterprise Postgres.

Structure of this document

This document is structured as follows:

Chapter 1 New Features and Improvements

Explains the new features and improvements in this version.

Chapter 2 Compatibility Information

Provides information regarding compatibility.

Chapter 3 Program Updates

Explains updates incorporated in this version.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

```
Edition 1.2: August 2023
Edition 1.1: June 2023
Edition 1.0: April 2023
```

Copyright

Copyright 2015-2023 Fujitsu Limited

Revision History

Revision	Location	Version
Added an incompatible article about migrating applications.	2.2.1 Changing the OID of the Data Type (NCHAR type) that Handles National Characters	Edition 1.2
Feature names have been improved. Before modification : Confidential management support After modification : Confidentiality management Before modification : Cloud Key Management Service Integration After modification : Cloud-based Key Management Service Integration	Table 1.1 New features and improvements	Edition 1.1
	1.1.1 Security	
	1.1.1.1 Confidentiality Management	
	1.1.1.2 Cloud-based Key Management Service Integration	

Contents

Chapter 1 New Features and Improvements....1
1.1 Features Added in 15....1
1.1.1 Security....1
1.1.1.1 Confidentiality Management....1
1.1.1.2 Cloud-based Key Management Service Integration....1
1.1.2 Auditing....2
1.1.2.1 pgaudit Improvements....2
1.1.3 OSS....2
1.1.3.1 PostgreSQL Rebase....2
1.1.3.2 OSS Updates Provided....2
1.1.4 Platform Enhancement....2
1.1.4.1 Additional Operating System Support for Server Feature....2
1.1.4.2 Additional Operating System Support for Client Feature....3
1.1.4.3 Additional Operating System Support for Server Assistant Feature....3
1.1.5 Application Development....3
1.1.5.1 Addition of JDKs and JREs on which the JDBC Driver can Work....3
1.1.6 Miscellaneous (LD_LIBRARY_PATH Improvements)....3
1.1.6.1 Additonal the DT_RUNPATH Attribute....3
1.1.6.2 About Building Applications and Extensions....4

Chapter 2 Compatibility Information....5
2.1 Installation/Setup Incompatibility....5
2.1.1 Removing Operating System Support for Client Feature....5
2.1.2 Removing Operating System Support for Client Feature....6
2.1.3 Removing Operating System Support for Server Feature....6
2.1.4 Changing kernel parameter settings when an instance is created with WebAdmin....6
2.1.5 Removing Operating System Support for Client Feature....7
2.1.6 Removing Operating System Support for Server Feature....7
2.1.7 Changing the Way OSS is Set Up....8
2.1.8 Modifying Pgpool-II Installation Handling....8
2.1.9 Changing Core and Log File Paths when Instance is Created with WebAdmin....8
2.1.10 Renaming WebAdmin Services....9
2.1.11 Changing the Features Targeted for Installation in a 64-bit Environment....9
2.1.12 Changing the Access Permissions of the Windows Client Installation Folder....9
2.1.13 Changing the WebAdmin Installation Method....9
2.2 Application Migration Incompatibility....10
2.2.1 Changing the OID of the Data Type (NCHAR type) that Handles National Characters....10
2.2.2 Changed the Specification Method of the Application Connection Switch Feature....11
2.2.3 Changes in ecpg/ecobpg Handling of Zero-Length Unicode Identifiers....11
2.2.4 Changing the display result when data masking is applied to NaN, infinity, -infinity....11
2.2.5 Changing the Valid Range of Identifiers Defined by the DECLARE STATEMENT statement....11
2.2.6 Changing Precompile Results....12
2.2.7 Changing the Trigger Replacement Process....12
2.2.8 Change the "elif" Statement Behavior in ecobpg....12
2.2.9 Changed to Error when Running an Operator or Function that Returns non Data Types for Masking Type....13
2.2.10 Changing the Method of Specifying the Application Connection Switch Feature....14
2.2.11 PostgreSQL Compatibility of Embedded SQL Applications in C and COBOL....14
2.2.12 Changing Vertical Clustered Index (VCI)....15
2.2.13 Changing how to Use the Features Compatible with Oracle Databases....15
2.3 Operation Migration Incompatibility....15
2.3.1 Abolition of Message Numbers....16
2.3.2 Adding the key_name Column to the View pgx_tde_master_key....17
2.3.3 Changing the Output of the Status Mode of the cm_ctl Command....17
2.3.4 Rename column "master_pid" in pgx_loader_state to "leader_pid"....18
2.3.5 Adding a Message to Output when the Database Server watchdog detects that the Connection Manager is down....18

2.3.6 Change the Error Information when the Connection Manager re-executes SQL on the Failed Connection 19
2.3.7 Changing the Value of the Category Column in the pg_settings view 19
2.3.8 Changing pgx_stat_lwlock of the Statistics View 20
2.3.9 Changing the Behavior of pgx_rcvall 21
2.3.10 Mirroring Controller no Longer Retries to Monitor Database Processes when they are Detected as Down 21
2.3.11 Changing the Name and Parameter Name of the Mirroring Controller Post-Promote Command 22
2.3.12 Changing Mirroring Controller User Command Input Values 22
2.3.13 Changing the Maximum Number of Connections per Server 22
2.3.14 Changing the Encryption Settings Using the ALTER TABLESPACE Statement 23
2.3.15 Changing the Default Configuration of the Cluster System Using Database Multiplexing 23
2.3.16 Changing the Default Operation when mc_ctl Command Options are Omitted 23
2.3.17 Changing the Connection Settings when Mirroring Controller Connects to an Instance 24
2.3.18 Changing the Status Display of the Mirroring Controller Server 24
2.3.19 Changing the Operation when the synchronous_standby_names Parameter is Changed during Database Multiplexing Operation 25
2.3.20 Changing Masking Policy Definition for Unsupported Data Types 25
2.4 JDBC Driver Incompatibility 25
2.4.1 Changing the targetServerType Value 26
2.5 ODBC Driver Incompatibility 26
2.5.1 Cannot specify prefer-read for target_session_attrs 26
2.6 .NET Data Provider Incompatibility 27
2.6.1 Changing the Specification Method and Specified Values for the Application Connection Switch Feature 27
2.6.2 Changing the Behavior of the Application Connection Switch Feature 28
2.6.3 Behavior Change when Multiple Host Information is Specified in the Connection String 28
2.6.4 Changing the TargetServerType Value 28
2.6.5 Changing the Server Explorer View in Visual Studio Integration with .NET Data Provider 29
2.7 C Library (libpq) Migration Incompatibility 29
2.7.1 Changing when "prefer-read" is Specified for the target_session_attrs Parameter 29
2.8 oracle_fdw Incompatibility 30
2.8.1 Changing the Oracle Client Version 30
2.9 pgaudit Incompatibility 30
2.9.1 Changing to Output Extra NEW and OLD Values in the Audit Log when the Trigger Function Executes 30
2.10 WebAdmin Incompatibility 31
2.10.1 Cannot specify prefer-read for target_session_attrs 31
2.11 Connection Manager Incompatibility 31
2.11.1 Behavior change when "read-write" is specified for the target_session_attrs parameter 32

Chapter 3 Program Updates 33

Index 34

Chapter 1 New Features and Improvements

This chapter explains Fujitsu Enterprise Postgres new features and improvements added in this version.

Table 1.1 New features and improvements

Version and level	Classification	Feature	Provided in AE	Provided in SE
15	Security	Confidentiality Management	Y	N
		Cloud-based Key Management Service Integration	Y	Y
	Auditing	pgaudit Improvements	Y	N
	OSS	PostgreSQL Rebase	Y	Y
		OSS updates provided	Y	Y
	Platform enhancement	Additional Operating System Support for Server Feature	Y	Y
		Additional Operating System Support for Client Feature	Y	N
		Additional Operating System Support for Server Assistant Feature	Y	N
	Application Development	Add JDKs and JREs that can run JDBC drivers	Y	Y
	Miscellaneous (LD_LIBRARY_PATH Improvements)	Additonal the DT_RUNPATH Attribute	Y	Y
		About Building Applications and Extensions	Y	Y

1.1 Features Added in 15

This section explains new features and improvements in Fujitsu Enterprise Postgres 15.

1.1.1 Security

This section explains the new features and improvements related to security:

- Confidentiality Management
- Cloud-based Key Management Service Integration

1.1.1.1 Confidentiality Management

The confidentiality management function is provided to assist in the design and operation of access control for databases. This reduces the man-hours required to manage complicated access control.

See

Refer to the Security Operation Guide for more information.

1.1.1.2 Cloud-based Key Management Service Integration

Cloud-based key management service integration is now available. Plug-in mechanisms for calling system service communication adapters in the cloud and data encryption key sharing are now supported.

See

Refer to "Using Transparent Data Encryption with Key Management Systems as Keystores" in the Operation Guide for more information,

1.1.2 Auditing

This section describes new features and improvements related to auditing.

- pgaudit Improvements

1.1.2.1 pgaudit Improvements

The audit_log_disconnections parameter has been added.

See

Refer to Security Operations Guide for more information,

1.1.3 OSS

This section explains the new feature related to OSS:

- PostgreSQL Rebase
- OSS Updates Provided

1.1.3.1 PostgreSQL Rebase

The PostgreSQL version that Fujitsu Enterprise Postgres is based on is 15.0.

See

Refer to "PostgreSQL Version Used for Fujitsu Enterprise Postgres" in the Installation and Setup Guide for Server for more information.

1.1.3.2 OSS Updates Provided

The OSS provided by Fujitsu Enterprise Postgres has been updated.

See

Refer to "OSS Supported by Fujitsu Enterprise Postgres" in the General Description for details.

1.1.4 Platform Enhancement

This section explains the new features related to platform enhancement:

- Additional operating system support for server
- Additional operating system support for client
- Additional operating system support for server assistant

1.1.4.1 Additional Operating System Support for Server Feature

The following additional operating system is supported:

- RHEL9

See

Refer to "Required Operating System" in the Installation and Setup Guide for Server for more information,

1.1.4.2 Additional Operating System Support for Client Feature

The following additional operating system is supported:

- RHEL9

See

Refer to "Required Operating System" in the Installation and Setup Guide for Server for more information,

1.1.4.3 Additional Operating System Support for Server Assistant Feature

The following additional operating system is supported:

- RHEL9

See

Refer to "Required Operating System" in the Installation and Setup Guide for Server Assistant for more information,

1.1.5 Application Development

This section describes new features related to application development.

- Addition of JDKs and JREs on which the JDBC Driver can Work

1.1.5.1 Addition of JDKs and JREs on which the JDBC Driver can Work

The following JDKs and JREs have been added to the list of supported JDKs and JREs.

- JDK 17 and JRE 17

See

Refer to "Related Software" in the Installation Guide for Client for details,

1.1.6 Miscellaneous (LD_LIBRARY_PATH Improvements)

Describes the LD_LIBRARY_PATH improvements.

- Additonal the DT_RUNPATH Attribute
- About Building Applications and Extensions

1.1.6.1 Additonal the DT_RUNPATH Attribute

Add DT_RUNPATH attribute to Fujitsu Enterprise Postgres programs.

This allows Fujitsu Enterprise Postgres programs to run without setting the environment variable LD_LIBRARY_PATH to the path to the installation directory/lib for Fujitsu Enterprise Postgres features.

1.1.6.2 About Building Applications and Extensions

If you are building an application that makes use of the shared libraries shipped with Fujitsu Enterprise Postgres, or an extension module for Fujitsu Enterprise Postgres, ensure that the DT_RUNPATH attribute of the program is set to the path to the directory that contains those libraries.

This allows programs to run without setting the environment variable LD_LIBRARY_PATH to the path to the Fujitsu Enterprise Postgres installation directory/lib.

See

Refer to about setting the DT_RUNPATH attribute, see:

- "How to Build and Run an Application that Uses Shared Libraries" in the Application Development Guide
- "Build with PGXS" and "Build without PGXS" in "Setting Up and Removing OSS" in the Installation and Setup Guide for Server

You can also continue to set the environment variable LD_LIBRARY_PATH to the path to the Fujitsu Enterprise Postgres installation directory/lib, and run these programs without setting the DT_RUNPATH attribute.

See

For notes on setting the environment variable LD_LIBRARY_PATH, refer to "When DT_RUNPATH cannot be set" in "How to Build and Run an Application that Uses Shared Libraries" in the Application Development Guide.

Chapter 2 Compatibility Information

This chapter explains incompatible items and actions required when migrating from an earlier version to Fujitsu Enterprise Postgres 15. Check compatibility before migrating and take the appropriate action.

2.1 Installation/Setup Incompatibility

Item	Pre-migration version								
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14	14 SP1
Removing Operating System Support for Client Feature	Y	Y	Y	Y	Y	Y	N	N	N
Removing Operating System Support for Client Feature	N	N	Y	Y	Y	N	N	N	N
Removing Operating System Support for Server Feature	N	N	Y	Y	Y	N	N	N	N
Changing kernel parameter settings when an instance is created with WebAdmin	Y	Y	Y	Y	Y	N	N	N	N
Removing Operating System Support for Client Feature	Y	Y	Y	Y	N	N	N	N	N
Removing Operating System Support for Server Feature	Y	Y	Y	Y	N	N	N	N	N
Changing the way OSS is set up	Y	Y	Y	Y	N	N	N	N	N
Modifying Pgpool-II Installation Handling	Y	Y	Y	Y	N	N	N	N	N
Changing Core and Log File Paths when Instance is Created with WebAdmin	Y	Y	Y	Y	N	N	N	N	N
Renaming WebAdmin Services	Y	Y	Y	Y	N	N	N	N	N
Changing the features targeted for installation in a 64-bit environment	Y	Y	N	N	N	N	N	N	N
Changing the access permissions of the Windows client installation folder	Y	Y	N	N	N	N	N	N	N
Changing the WebAdmin installation method	Y	N	N	N	N	N	N	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.1.1 Removing Operating System Support for Client Feature

Incompatibility

In FUJITSU Enterprise Postgres 13 SP1 or later, the following operating systems have been removed.

- Windows Server(R) 2012
- Windows Server(R) 2012 R2

Action method

None.

2.1.2 Removing Operating System Support for Client Feature

Incompatibility

In FUJITSU Enterprise Postgres 13 or later, the following operating systems have been removed.

- SLES 12 SP4 or earlier

Action method

None.

2.1.3 Removing Operating System Support for Server Feature

Incompatibility

In FUJITSU Enterprise Postgres 13 or later, the following operating systems have been removed.

- SLES 12 SP4 or earlier

Action method

None.

2.1.4 Changing kernel parameter settings when an instance is created with WebAdmin

Incompatibility

For FUJITSU Enterprise Postgres 13 and later, changes kernel parameter settings for WebAdmin instance creation.

FUJITSU Enterprise Postgres 12 or earlier

Kernel Parameters	Value	Calculated Value
SHMMAX	If *currentValue* < *calculatedValue*, configure the calculated value	`((1800 + 270 * max_locks_per_transaction) * max_connections + (1800 + 270 * max_locks_per_transaction) * autovacuum_max_workers + (770 + 270 * max_locks_per_transaction) * max_prepared_transactions + (shared_buffer) + (16 * 1024 * 1024) + ( 770 * 1024)) * 1.05`

SHMALL	Specify *currentValue* + *calculatedValue*	`(SHMMAX / PAGESIZE) + 1`
		
		`PAGESIZE = 4K`

SEMMNI	Specify *currentValue* + *calculatedValue*	`ceil((max_connections + autovacuum_max_workers + 4) / 16)`
SEMMNS	Specify *currentValue* + *calculatedValue*	`ceil((max_connections + autovacuum_max_workers + 4) / 16) * 17`

FUJITSU Enterprise Postgres 13 or later

Kernel Parameters	Value	Calculated Value
SHMMAX	Do not change value	-
SHMALL	Do not change value	-
SEMMNI	Specify *currentValue* + *calculatedValue*	- For instances of FUJITSU Enterprise Postgres 9.5 to 11:

Kernel Parameters	Value	Calculated Value
		ceil((max_connections + autovacuum_max_workers + max_worker_processes + 5) / 16) - For Fujitsu Enterprise Postgres 12 and later instances: ceil((max_connections + autovacuum_max_workers + max_wal_senders + max_worker_processes + 5) / 16)
SEMMNS	Specify *currentValue* + *calculatedValue*	- For instances of FUJITSU Enterprise Postgres 9.5 to 11: ceil((max_connections + autovacuum_max_workers + max_worker_processes + 5) / 16) * 17 - For Fujitsu Enterprise Postgres 12 and later instances: ceil((max_connections + autovacuum_max_workers + max_wal_senders + max_worker_processes + 5) / 16) * 17

Action method

None.

2.1.5 Removing Operating System Support for Client Feature

Incompatibility

In FUJITSU Enterprise Postgres 12 or later, the following operating systems have been removed.

- Windows(R) 7
- Windows Server(R) 2008
- RHEL6

Also, the 32 bit Linux client can no longer be installed because RHEL 6 has been removed from the operating system.

Action method

None.

2.1.6 Removing Operating System Support for Server Feature

Incompatibility

In FUJITSU Enterprise Postgres 12 or later, the following operating systems have been removed.

- RHEL6

Action method

None.

2.1.7 Changing the Way OSS is Set Up

Incompatibility

FUJITSU Enterprise Postgres 12 or later do not place OSS extension modules in the executable directory. The OSS extension modules must be placed in the executable directory when you set up OSS.

Refer to "Setting Up and Removing OSS" in the Installation and Setup Guide for Server for details.

Action method

None.

2.1.8 Modifying Pgpool-II Installation Handling

Incompatibility

For FUJITSU Enterprise Postgres 12 or later, Pgpool-II is not automatically installed when you install the server. Therefore, if you want to take advantage of Pgpool-II, install it separately from the server installation.

The extension modules required for the database server are shipped with the server program. You should set up Pgpool-II on the database server side, even if Pgpool-II is to be used on a different server than the database server.

Refer to "Setting Up and Removing OSS" in the Installation and Setup Guide for Server for details.

Action method

None.

2.1.9 Changing Core and Log File Paths when Instance is Created with WebAdmin

Incompatibility

In FUJITSU Enterprise Postgres 12 or later, change the core and log file paths when creating an instance in WebAdmin.

FUJITSU Enterprise Postgres 11 or earlier

Log File Path: /var/tmp/fsep_*version*/*instanceAdminUser_instanceNamePortNumber*/log

Core File Path: /var/tmp/fsep_*version*/*instanceAdminUser_instanceNamePortNumber*/core

version: product version_edition_architecture

［Example］

Log File Path: /var/tmp/fsep_110_AE_64/naomi_myinst27599/log

Core File Path: /var/tmp/fsep_110_AE_64/naomi_myinst27599/core

FUJITSU Enterprise Postgres 12

Log File Path: /var/tmp/fsep_*version*/*instanceAdminUser_instanceNamePortNumber*/log

Core File Path: /var/tmp/fsep_*version*/*instanceAdminUser_instanceNamePortNumber*/core

version: product version_WA_architecture

［Example］

Log File Path: /var/tmp/fsep_120_WA_64/naomi_myinst27599/core

Core File Path: /var/tmp/fsep_120_WA_64/naomi_myinst27599/log

Action method

None.

2.1.10 Renaming WebAdmin Services

Incompatibility

In FUJITSU Enterprise Postgres 12 or later, change the service name registered when you set up WebAdmin.

FUJITSU Enterprise Postgres 11 or earlier

fsep_*x*SP*z_edition*_64_WebAdmin_Port1

fsep_*x*SP*z_edition*_64_WebAdmin_Port2

FUJITSU Enterprise Postgres 12 or later

fsep_xSPz_WA_64_WebAdmin_Port1

fsep_xSPz_WA_64_WebAdmin_Port2

Action method

None.

2.1.11 Changing the Features Targeted for Installation in a 64-bit Environment

Incompatibility

32 bit Linux client can no longer be installed on RHEL7 in FUJITSU Enterprise Postgres 10 or later.

Note: 32 bit Linux client can install on RHEL6 in FUJITSU Enterprise Postgres 11 or earlier.

Action method

None.

2.1.12 Changing the Access Permissions of the Windows Client Installation Folder

Incompatibility

When changed for install folder the Windows client from default, the access permissions of the installed files and folder changes to read and execute.

The above changes apply to non-administrators.

This incompatibility occurs with all supported Windows client operating systems. Refer to "Required Operating System" in the Installation and Setup Guide for Client for information on the operating environment.

Action method

This change was made simply to enhance security, therefore, no action is normally required.

However, if existing access permissions are required, this change can be reverted by running *installDir*\setup\revert_cacls.bat as an administrator after installation.

2.1.13 Changing the WebAdmin Installation Method

Incompatibility

In FUJITSU Enterprise Postgres 9.6 or later, WebAdmin is not installed automatically during server installation. Therefore, install WebAdmin separately.

Action method

None.

2.2 Application Migration Incompatibility

Item	Pre-migration version								
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14	14 SP1
Changing the OID of the Data Type (NCHAR type) that Handles National Characters	Y	Y	Y	Y	Y	Y	Y	Y	Y
Changed the Specification Method of the Application Connection Switch Feature	Y	Y	Y	Y	Y	Y	Y	Y	Y
Changes in ecpg/ecobpg Handling of Zero-Length Unicode Identifiers	Y	Y	Y	Y	Y	Y	Y	Y	Y
Changing the display result when data masking is applied to NaN, infinity, -infinity	Y	Y	Y	Y	Y	Y	Y	N	N
Changing the Valid Range of Identifiers Defined by the DECLARE STATEMENT statement	Y	Y	Y	Y	Y	Y	Y	N	N
Changing Precompile Results	Y	Y	Y	Y	Y	Y	Y	N	N
Changing the Trigger Replacement Process	Y	Y	Y	Y	Y	Y	Y	N	N
Change the "elif" Statement Behavior in ecobpg	Y	Y	Y	Y	Y	N	N	N	N
Changed to Error when Running an Operator or Function that Returns non Data Types for Masking Type	Y	Y	Y	Y	N	N	N	N	N
Changing the method of specifying the application connection switch feature	Y	Y	N	N	N	N	N	N	N
PostgreSQL compatibility of embedded SQL applications in C and COBOL	Y	Y	N	N	N	N	N	N	N
Changing Vertical Clustered Index (VCI)	Y	Y	N	N	N	N	N	N	N
Changing how to use the features compatible with Oracle databases	Y	N	N	N	N	N	N	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.2.1 Changing the OID of the Data Type (NCHAR type) that Handles National Characters

Incompatible

In Fujitsu Enterprise Postgres 15, OIDs for national character data types (NCHAR types) have changed.

Action method

If you are using a national character data type (NCHAR type), recompile the application and run it with Fujistu Enterprise Postgres 15 or later clients.

2.2.2 Changed the Specification Method of the Application Connection Switch Feature

Incompatible

In Fujitsu Enterprise Postgres 15, the name of the "target server" specified by the application connection switch feature will be changed.

Action method

For applications that specify a target server as the application connection switch feature, change the target server according to the table below, recompile, and then execute using a client of Fujistu Enterprise Postgres 15 or later.

Table 2.1 Specified name of the target server.

Target of use	Before modification	After modification
Leverage .NET Data Provider	TargetServerType	TargetSessionAttributes

2.2.3 Changes in ecpg/ecobpg Handling of Zero-Length Unicode Identifiers

Incompatible

In Fujitsu Enterprise Postgres 15, if the SQL statement contains a zero-length Unicode identifier, Change precompilation to fail.

Action method

None.

2.2.4 Changing the display result when data masking is applied to NaN, infinity, -infinity

Incompatibility

In FUJITSU Enterprise Postgres 14, the display result when data masking is applied to NaN, infinity, and -infinity will be changed.

FUJITSU Enterprise Postgres 13 SP1 or earlier

If the float type NaN, infinity, and -infinity are partially masking with (9,1,2), the following will be displayed.

```
NaN : 99
Infinity : 99
-Infinity : 99
```

FUJITSU Enterprise Postgres 14 or later

If the float type NaN, infinity, and -infinity are partially masking with (9,1,2), the following will be displayed.

```
NaN : NaN
Infinity : Infinity
-Infinity : -Infinity
```

Action method

If the application is analyzing the SQL masking output result, please consider the non-numeric output result and correct it.

2.2.5 Changing the Valid Range of Identifiers Defined by the DECLARE STATEMENT statement

Incompatibility

In FUJITSU Enterprise Postgres 14 will change the valid range of identifiers defined by a DECLARE STATEMENT statement in ecpg/ecobpg.

FUJITSU Enterprise Postgres 13 SP1 or earlier

The valid range is now per process.

FUJITSU Enterprise Postgres 14 or later

The valid range is now per file.

Action method

None.

2.2.6 Changing Precompile Results

Incompatibility

In FUJITSU Enterprise Postgres 14 removed the ECPGdeclare/ECPGopen function. Therefore, results precompiled from earlier versions of FUJITSU Enterprise Postgres will not be available in FUJITSU Enterprise Postgres 14.

Action method

Rebuild the application.

2.2.7 Changing the Trigger Replacement Process

Incompatibility

In FUJITSU Enterprise Postgres 14 will change restricted triggers to not be supported by replace operations (OR REPLACE).

FUJITSU Enterprise Postgres 13 SP1 or earlier

You can replace a constraint trigger.

[Example]

```
=# CREATE OR REPLACE CONSTRAINT TRIGGER my_constraint_trigger AFTER DELETE ON my_table
-# FOR EACH ROW
-# EXECUTE PROCEDURE funcA();
CREATE TRIGGER
```

FUJITSU Enterprise Postgres 14 or later

It does not support replacing a constraint trigger.

[Example]

```
=# CREATE OR REPLACE CONSTRAINT TRIGGER my_constraint_trigger AFTER DELETE ON my_table
-# FOR EACH ROW
-# EXECUTE PROCEDURE funcA();
ERROR:  CREATE OR REPLACE CONSTRAINT TRIGGER is not supported
```

Action method

None.

2.2.8 Change the "elif" Statement Behavior in ecobpg

Incompatibility

The "elif" statement now works correctly with FUJITSU Enterprise Postgres 13.

FUJITSU Enterprise Postgres 12 or earlier

The decision of the "elif" statement is made whether or not the "ifdef/ifndef" condition is met.

[Example]

```
000000     EXEC SQL ifdef DEF1 END-EXEC.
000000     Operation (1).
000000     EXEC SQL elif DEF2 END-EXEC.
000000     Operation (2).
000000     EXEC SQL else END-EXEC.
000000     Operation (3).
```

If "DEF1" and "DEF2" are true, both "Operation (1)" and "Operation (2)" are executed.

FUJITSU Enterprise Postgres 13 or later

If the condition "ifdef/ifndef" is met, the judgment of the "elif" statement is not enforced.

[Example]

```
000000     EXEC SQL ifdef DEF1 END-EXEC.
000000     Operation (1).
000000     EXEC SQL elif DEF2 END-EXEC.
000000     Operation (2).
000000     EXEC SQL else END-EXEC.
000000     Operation (3).
```

If "DEF1" and "DEF2" are true, only "Operation (1)" is executed.

Action method

Check your existing application and fix the elif statement to work correctly.

2.2.9 Changed to Error when Running an Operator or Function that Returns non Data Types for Masking Type

Incompatibility

In FUJITSU Enterprise Postgres 12, changed to error when running an operator or function that returns non Data Types for Masking type.

FUJITSU Enterprise Postgres 11 or earlier

In the following cases, the operator or function in 3) may be executed without masking.

This is an incorrect result because the data containing the masking column is performed without masking.

1) Create a Masking policy. and

2) Execute a SELECT statement. and

3) Execute an operator or function in the SELECT statement of 2). and

4) The operator or function argument of 3) includes a subquery. and

5) The operator or function argument of 3) contains the column to be protected created in 1). and

6) The operator or function of 3) returns a type not listed in "Data Types for Masking" in "Operation Guide".

FUJITSU Enterprise Postgres 12 or later

Operator or function results of 3) in the following error.

```
ERROR:  The output data type is incompatible with the confidential policy.
HINT:  Data type of the result value(s) produced by expression/function using confidential columns
is not supported by Data masking module. Consider removing confidential columns from such
expressions/functions.
```

This is correct because the operator or function returns a data type not listed in "Data Types for Masking" in "Operation Guide".

Action method

Do one of the following to ensure that the results are identical to those of FUJITSU Enterprise Postgres 11 and earlier:

1) Modify the Masking policy to prevent masking from being performed for the user executing the SQL.

2) Modify SQL to not use operators or functions that return types not listed in "Data Types for Masking" in "Operation Guide".

2.2.10 Changing the Method of Specifying the Application Connection Switch Feature

Incompatibility

The target server specified using the application connection switch feature is changed to the same name as that of PostgreSQL.

Action method

Use the FUJITSU Enterprise Postgres 10 or later client to execute applications that have the target server specified as the application connection switch feature after changing the target server and specified value as listed below and recompiling.

Specified name of the target server

Usage target	Before correction	After correction
JDBC driver	No change	
ODBC driver	TargetServer	target_session_attrs
.NET Data Provider	target_server	TargetServerType
Connection service file	target_server	target_session_attrs
Library (libpq) for C	target_server Environment variable (PGXTARGETSERVER)	target_session_attrs Environment variable (PGTARGETSESSIONATTRS)
Embedded SQL	target_server	target_session_attrs
psql	Environment variable (PGXTARGETSERVER)	Environment variable (PGTARGETSESSIONATTRS)

Specified value for the target server

Server selection order	JDBC driver	.NET Data Provider		Other driver	
		Before correction	After correction	Before correction	After correction
Primary server	No change	primary	master	primary	read-write
Standby server	No change	-	slave	-	-
Priority given to a standby server	No change	prefer_standby	preferSlave	prefer_standby	prefer-read
Any	No change	-	any	-	any

2.2.11 PostgreSQL Compatibility of Embedded SQL Applications in C and COBOL

Incompatibility

Embedded SQL applications in C and COBOL that were compiled with FUJITSU Enterprise Postgres 9.6 or earlier cannot be used in the FUJITSU Enterprise Postgres 10 client.

Action method

Use the FUJITSU Enterprise Postgres 10 or later client to execute the applications only after recompiling them with the FUJITSU Enterprise Postgres 10 or later client.

2.2.12 Changing Vertical Clustered Index (VCI)

Incompatibility

The functions below have been added to the functions for which VCI is not used.

Type	Function
Date/time functions	age(timestamp) current_date current_timestamp current_time localtime localtimestamp
Session information functions	current_user current_role

Action method

To use VCI, specify the second argument when using the age function, and specify other functions as subqueries.

[Example]

Before: select age(column A), current_date from table

After: select age(column A, **now()**), **(select current_date)** from table

2.2.13 Changing how to Use the Features Compatible with Oracle Databases

Incompatibility

In FUJITSU Enterprise Postgres 9.6 or later, to use the features compatible with Oracle databases, create a new instance and execute the following command for the "postgres" and "template1" databases:

CREATE EXTENSION oracle_compatible

Action method

None.

2.3 Operation Migration Incompatibility

Item	Pre-migration version									
	9.5	9.6	10	11	12	13	12 SP1	13 SP1	14	14 SP1
Abolition of Message Numbers	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Adding the key_name Column to the View pgx_tde_master_key	N	N	N	N	N	N	N	N	N	Y
Changing the Output of the Status Mode of the cm_ctl Command	N	N	N	N	Y	Y	Y	Y	N	N
Rename column "master_pid" in pgx_loader_state to "leader_pid"	N	N	Y	Y	Y	Y	Y	Y	N	N

Item	Pre-migration version									
	9.5	9.6	10	11	12	13	12 SP1	13 SP1	14	14 SP1
Adding a Message to Output when the Database Server watchdog detects that the Connection Manager is down	N	N	N	N	N	N	Y	N	N	N
Change the Error Information when the Connection Manager re-executes SQL on the Failed Connection	N	N	N	N	N	N	Y	N	N	N
Changing the Value of the Category Column in the pg_settings view	N	Y	Y	Y	Y	N	N	N	N	N
Changing pgx_stat_lwlock of the Statistics View	N	Y	Y	Y	Y	N	N	N	N	N
Changing the Behavior of pgx_rcvall	Y	Y	Y	Y	Y	N	N	N	N	N
Mirroring Controller no longer retries to monitor database processes when they are detected as down	Y	N	Y	Y	N	N	N	N	N	N
Changing the Name and Parameter Name of the Mirroring Controller Post-Promote Command	N	N	Y	Y	N	N	N	N	N	N
Changing Mirroring Controller User Command Input Values	N	N	Y	Y	N	N	N	N	N	N
Changing the maximum number of connections per server	Y	Y	N	N	N	N	N	N	N	N
Changing the encryption settings using the ALTER TABLESPACE statement	Y	Y	N	N	N	N	N	N	N	N
Changing the default configuration of the cluster system using database multiplexing	Y	N	N	N	N	N	N	N	N	N
Changing the default operation when mc_ctl command options are omitted	Y	N	N	N	N	N	N	N	N	N
Changing the connection settings when Mirroring Controller connects to an instance	Y	N	N	N	N	N	N	N	N	N
Changing the status display of the Mirroring Controller server	Y	N	N	N	N	N	N	N	N	N
Changing the operation when the synchronous_standby_names parameter is changed during database multiplexing operation	Y	N	N	N	N	N	N	N	N	N
Changing masking policy definition for unsupported data types	Y	N	N	N	N	N	N	N	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.3.1 Abolition of Message Numbers

Incompatibility

In Fujitsu Enterprise Postgres 15, the message number output at the end of the message is abolished.

Message numbers are output for messages output by Mirroring Controller.

For FUJITSU Enterprise Postgres 14 SP1 or earlier

The message number was printed at the end of the message.

[example]

```
3D000: 2017-07-10 19:41:05 JST[13899]: [1-1] user=fepuser,db=fep,remote=127.0.0.1(51902)
app=[unknown] FATAL: database "fep" does not exist (10571)
```

For Fujitsu Enterprise Postgres 15

No message number is output at the end of the message.

[example]

```
3D000: 2023-04-10 19:41:05 JST [13899]: [1-1] user = fepuser,db = fep,remote = 127.0.0.1(51902)
app = [unknown] FATAL: database "fep" does not exist
```

Action method

None.

2.3.2 Adding the key_name Column to the View pgx_tde_master_key

Incompatibility

In Fujitsu Enterprise Postgres 15, add a key_name column to the view pgx_tde_master_key.

Action method

None.

2.3.3 Changing the Output of the Status Mode of the cm_ctl Command

Incompatibility

Changes the display of the output of the status mode of the cm_ctl command.

(If it has been modified by the P number PH21029, and you have applied an urgent fix that includes it, FUJITSU Enterprise Postgres 14 will work.)

When the cm_ctl command was executed in status mode, there was omission of the output of the header "pid" displayed in "application_information" which outputs the information of the application.

Correct the following in PH21029.

- Add ':' to 'application_information' to output application information
  Output "application_information:"
- Add the header "pid" output to "application_information"
  Outputs "pid" whose header information is missing.
- Changes the number of digits in the numeric portion of connected_time
  Change the display start position of the date and time connected to the conmgr process to correct the gap between the start position of the header and the numeric part. (5 digits backward)

This fix causes the following incompatibilities when running the cm_ctl command in status mode:

- Add ":" to the display of "application_information"
- Display the header "pid" in "application_information"
- Change the output start position of the "connected_time" header of "application_information"
- Change the start of the date and time output of "connected_time" in "application_information" to 5 digits later.

FUJITSU Enterprise Postgres 13 SP1 or earlier

```
application_information
addr         port  connected_time
10.xxx.x.xx  99999 21655 2021-10-20 09:18:51
```

FUJITSU Enterprise Postgres 14

```
application_information:
addr         port  pid       connected_time
10.xxx.x.xx  99999 21655     2021-10-20 09:18:51
```

When analyzing the output of the cm_ctl command in status mode in a batch or shell script, it may not work correctly if the header is referenced and the third is "connected_time".

For example, when identifying a row of data (numeric part), a string up to the header "addr", "port", and "connected_time" one row before is searched for and identified.

Action method

If you are using a batch or shell script to parse the output of the cm_ctl command in status mode, modify it to take into account the number of digits in the header and numeric part of the output.

2.3.4 Rename column "master_pid" in pgx_loader_state to "leader_pid"

Incompatibility

In FUJITSU Enterprise Postgres 14 renames column "master_pid" to "leader_pid" in the pgx_loade_state table.

Action method

None.

2.3.5 Adding a Message to Output when the Database Server watchdog detects that the Connection Manager is down

Incompatibility

In FUJITSU Enterprise Postgres 12 SP1, when using the Connection Manager, if the database server watchdog detects that the Connection Manager is down, it will output a message to the database server.

Incompatibilities may occur if:

1) The application server is using the Connection Manager. and

2) The database server is running the watchdog process. and

3) 1) Application server or Connection Manager goes down.

FUJITSU Enterprise Postgres 12 or earlier

If the database server watchdog detects that the Connection Manager is down, the following message is not output.

```
WARNING:  watchdog: error in heartbeat connection (20331): host=xxx.xxx.xxx.xxx port=xxxxx
pid=xxxxx
```

FUJITSU Enterprise Postgres 12 SP1

If the database server watchdog detects that the Connection Manager is down, the following message may be output.

```
WARNING:  watchdog: error in heartbeat connection (20331): host=xxx.xxx.xxx.xxx port=xxxxx
pid=xxxxx
```

FUJITSU Enterprise Postgres 13 or later

If the database server watchdog detects that the Connection Manager is down, the following message is not output.

```
WARNING:  watchdog: error in heartbeat connection (20331): host=xxx.xxx.xxx.xxx port=xxxxx
pid=xxxxx
```

Action method

If you are monitoring the database log for watchdog messages, change the monitoring setting to one that takes into account the possible output of these messages.

2.3.6 Change the Error Information when the Connection Manager re-executes SQL on the Failed Connection

Incompatibility

In FUJITSU Enterprise Postgres 12 SP1 allows an application using the Connection Manager to change the error information when attempting to execute SQL again on a connection that the Connection Manager has determined to be in error due to a database error.

The changed error information is included in the error presented in "Errors when an Application Connection Switch Occurs and Corresponding Actions" for each client driver in "Application Development Guide".

Incompatibilities may occur if the following conditions are met:

1) The application uses one of the following drivers. and

-libpq (C Library)

-ECPG (Embedded SQL in C)

2) You are using the Connection Manager. and

3) The database server to which the application in 1) was connected goes down. and

4) The Connection Manager notifies the application in 1) of an error when the database is down in 3). and

5) The application in 1) does not disconnect from the database server that is down, but executes SQL using the connection.

FUJITSU Enterprise Postgres 12 or earlier

The error information returned depends on the actual database server error condition.

FUJITSU Enterprise Postgres 12 SP1

The following error information is returned:

- For libpq (C Library): CONNECTION_BAD (PQstatus () return value)
- For ECPG (Embedded SQL in C): 57P02 (SQLSTATE return value)

FUJITSU Enterprise Postgres 13 or later

The error information returned depends on the actual database server error condition.

Action method

Consider that the above error may be returned during the SQL error determination process.

Also, if you encounter an error about switching application destinations as described in "Errors when an Application Connection Switch Occurs and Corresponding Actions" for each client driver in "Application Development Guide", you should explicitly disconnect and reconnect or rerun the application.

2.3.7 Changing the Value of the Category Column in the pg_settings view

Incompatibility

For FUJITSU Enterprise Postgres 13, change the value of the category column in the pg_settings view.

FUJITSU Enterprise Postgres 12 or earlier

Original value	wrong value
Preset Options	Fujitsu Enterprise Postgres Parameters
Customized Options	Preset Options
Developer Options	Customized Options

This is an incorrect result because it is different from the original value.

FUJITSU Enterprise Postgres 13 or later

The correct value is the category column in the pg_settings view.

Action method

Replaces the category column in the pg_settings view with the values before migration, so that the results are the same as before migration.

2.3.8 Changing pgx_stat_lwlock of the Statistics View

Incompatibility

In FUJITSU Enterprise Postgres 13, change the wait event name displayed in the lwlock_name column of the statistics view pgx_stat_lwlock.

Wait Event Name

FUJITSU Enterprise Postgres 12 or earlier	FUJITSU Enterprise Postgres 13
clog (*1)	XactBuffer
commit_timestamp (*1)	CommitTSBuffer
subtrans (*1)	SubtransBuffer
multixact_offset (*1)	MultiXactOffsetBuffer
multixact_member (*1)	MultiXactMemberBuffer
async (*1)	NotifyBuffer
oldserxid (*1)	SerialBuffer
wal_insert (*1)	WALInsert
buffer_content (*1)	BufferContent
buffer_io (*1)	BufferIO
replication_origin (*1)	ReplicationOriginState
replication_slot_io (*1)	ReplicationSlotIO
proc (*1)	LockFastPath
buffer_mapping (*1)	BufferMapping
lock_manager (*1)	LockManager
predicate_lock_manager (*1)	PredicateLockManager
parallel_hash_join (*3)	ParallelHashJoin
parallel_query_dsa (*2)	ParallelQueryDSA
session_dsa (*3)	PerSessionDSA
session_record_table (*3)	PerSessionRecordType

FUJITSU Enterprise Postgres 12 or earlier	FUJITSU Enterprise Postgres 13
session_typmod_table (*3)	PerSessionRecordTypmod
shared_tuplestore (*3)	SharedTupleStore
tbm (*2)	SharedTidBitmap
parallel_append (*3)	ParallelAppend
serializable_xact (*4)	PerXactPredicateList
shared_mcxt (*4)	SharedMcxt
meta_cache_map (*4)	MetaCacheMap
global_metacache (*4)	GlobalCatcache
cached_buf_tranche_id (*4)	CachedBufTranche

*1) Events added in FUJITSU Enterprise Postgres 9.6.

*2) Events added in FUJITSU Enterprise Postgres 10.

*3) Events added in FUJITSU Enterprise Postgres 11.

*4) Events added in FUJITSU Enterprise Postgres 12.

Action method

None.

2.3.9 Changing the Behavior of pgx_rcvall

Incompatibility

In FUJITSU Enterprise Postgres 13, change the pgx_rcvall command to fail if the -e option of the pgx_rcvall command specifies a future time or if the -n option specifies a list appointment that does not exist.

FUJITSU Enterprise Postgres 12 or earlier

All archived WALs are applied for recovery if the -e option of the pgx_rcvall command specifies a time in the future, or if the -n option specifies a list appointment that does not exist.

FUJITSU Enterprise Postgres 13 or later

The pgx_rcvall command fails if the -e option of the pgx_rcvall command specifies a time in the future, or if the -n option specifies a list appointment that does not exist.

Action method

Specify recovery objectives correctly, if necessary.

2.3.10 Mirroring Controller no Longer Retries to Monitor Database Processes when they are Detected as Down

Incompatibility

For the FUJITSU Enterprise Postgres Mirroring Controller, change the heartbeat monitoring of the database process so that it does not retry monitoring when it detects down.

Action method

None.

2.3.11 Changing the Name and Parameter Name of the Mirroring Controller Post-Promote Command

Incompatibility

In the FUJITSU Enterprise Postgres 12 Mirroring Controller, change the name of the post-promote command, which is the state transition command, and the parameter name in the server configuration file that specifies the post-promote command.

FUJITSU Enterprise Postgres 11 or earlier

- Command Name

  post-promote command

- The parameter name in the server configuration file that specifies the post-promote command

  post_promote_command

FUJITSU Enterprise Postgres 12 or later

- Command Name

  post-switch command

- The parameter name in the server configuration file that specifies the post-promote command

  post_switch_command

Action method

The post_promote_command parameter in the server configuration file continues to be available in FUJITSU Enterprise Postgres 12 or later.If specified in the server configuration file, it acts as a post-switch command.You cannot specify the post_promote_command and post_switch_command parameters at the same time.

2.3.12 Changing Mirroring Controller User Command Input Values

Incompatibility

Arguments (Fixed value: primarycenter) have been added to the following user commands:

- Fencing command of the database server
- Arbitration command
- Post-switch command
- Pre-detach command
- Post-attach command

Action method

If you are checking the number of arguments in a user command, increase the number of arguments by one.

2.3.13 Changing the Maximum Number of Connections per Server

Incompatibility

The maximum number of connections per server is changed from 262,143 to 65,535.

Action method

None.

2.3.14 Changing the Encryption Settings Using the ALTER TABLESPACE Statement

Incompatibility

The ALTER TABLESPACE statement can be used to change the tablespace encryption settings if data is not stored in the tablespace.

Action method

None.

2.3.15 Changing the Default Configuration of the Cluster System Using Database Multiplexing

Incompatibility

In FUJITSU Enterprise Postgres 10, a split brain will not occur if heartbeat monitoring using an admin network times out, so the default is changed to a system configuration requiring an arbitration server acting as a third party.

FUJITSU Enterprise Postgres 9.6 or earlier

A cluster system comprises two database servers.

FUJITSU Enterprise Postgres 10 or later

By default, a cluster system comprises two database servers and an arbitration server.

Action method

If selecting the same system configuration as that in FUJITSU Enterprise Postgres 9.6 or earlier, set the parameters below in the server configuration file to perform automatic degradation unconditionally when a heartbeat abnormality is detected during OS/server heartbeat monitoring.

- Parameter: heartbeat_error_action
- Value: fallback

2.3.16 Changing the Default Operation when mc_ctl Command Options are Omitted

Incompatibility

In FUJITSU Enterprise Postgres 10, the default operation when the mc_ctl command options below are omitted has been changed:

- The -f option during start mode
- The -w option during start mode

FUJITSU Enterprise Postgres 9.6 or earlier

- If the -f option is omitted, automatic switching and disconnection immediately after the startup of Mirroring Controller will not be enabled.
- If the -w option is omitted, the system will not wait for operations to finish.

FUJITSU Enterprise Postgres 10 or later

- Even if the -f option is omitted, automatic switching and disconnection immediately after the startup of Mirroring Controller will be enabled.
- Even if the -w option is omitted, the system will wait for operations to finish.

Action method

If selecting the same operation as that in FUJITSU Enterprise Postgres 9.6 or earlier, take the action below when executing start mode of the mc_ctl command.

- If the -f option is omitted, specify the -F option.
- If the -w option is omitted, specify the -W option.

2.3.17 Changing the Connection Settings when Mirroring Controller Connects to an Instance

Incompatibility

In FUJITSU Enterprise Postgres 10, Mirroring Controller changes the connection setting below when connecting to an instance to detect failure of each database element.

- Application name

FUJITSU Enterprise Postgres 9.6 or earlier

- The application name is an empty string.

FUJITSU Enterprise Postgres 10 or later

- The application name is 'mc_agent'.

Action method

There is no method for changing the application name.

Therefore, if there is an application that identifies a database connection session of Mirroring Controller with the application name being an empty string, modify the process so that identification is performed using 'mc_agent'.

2.3.18 Changing the Status Display of the Mirroring Controller Server

Incompatibility

In FUJITSU Enterprise Postgres 10, the condition for displaying an abnormality for the process (WAL receive process) that receives transaction logs has been changed in regard to the server status display performed in status mode of the mc_ctl command of Mirroring Controller.

- Role of the applicable server (host_role): standby (standby)
- Display item: DBMS process status (db_proc_status)
- Display content: abnormal (abnormal process name)
  The applicability for incompatibility is determined based on whether 'wal_receiver' is included in the abnormality process name.

FUJITSU Enterprise Postgres 9.6 or earlier

Displays an abnormality when Mirroring Controller detects that the WAL send process has stopped.

FUJITSU Enterprise Postgres 10 or later

Displays an abnormality when Mirroring Controller detects that a streaming replication connection has not been established with the primary server using the WAL receive process.

Action method

If there is an application that determines the WAL receive process status from the display content retrieved using status mode of the mc_ctl command, modify as below to determine at the same level of accuracy as previously. However, this is not recommended due to incompatibilities accompanying the improvement in accuracy.

- If 'wal_sender' is included in the abnormal process name displayed in the 'abnormal' db_proc_status item of the 'primary' host_role, it is determined that an abnormality has occurred in the WAL receive process.

- If 'wal_sender' is not included in the abnormal process name displayed in the 'abnormal' db_proc_status item of the 'primary' host_role, it is determined that an abnormality has not occurred in the WAL receive process.

2.3.19 Changing the Operation when the synchronous_standby_names Parameter is Changed during Database Multiplexing Operation

Incompatibility

In FUJITSU Enterprise Postgres 9.6 or earlier, Mirroring Controller periodically accesses the database to retrieve the synchronous_standby_names parameter value using the SHOW command, and automatically recovers if changes are detected, in case the user accidentally changes the synchronous_standby_names parameter in postgresql.conf during database multiplexing mode. However, because this process uses the CPU of the database server for redundancy and SQL statements are executed with high frequency, these processes are not executed by default in FUJITSU Enterprise Postgres 10.

FUJITSU Enterprise Postgres 9.6 or earlier

Mirroring Controller checks if the synchronous_standby_names parameter in postgresql.conf has been mistakenly changed by the user.

FUJITSU Enterprise Postgres 10 or later

By default, Mirroring Controller does not check if the synchronous_standby_names parameter in postgresql.conf has been mistakenly changed by the user.

Action method

If selecting the same operation as that in FUJITSU Enterprise Postgres 9.6 or earlier, set the parameter below in the server definition file.

- Parameter: check_synchronous_standby_names_validation
- Value: 'on'

2.3.20 Changing Masking Policy Definition for Unsupported Data Types

Incompatibility

The data masking feature of FUJITSU Enterprise Postgres is updated so that an error occurs when a masking policy is defined for a column of an unsupported data type (array type or timestamp with timezone type).

FUJITSU Enterprise Postgres 9.5

The definition of a masking policy for a column of array type or timestamp with timezone type ends normally, but an error occurs when accessing a column of a table for which the policy is defined.

FUJITSU Enterprise Postgres 9.6 or later

An error occurs when defining a masking policy for a column of array type or timestamp with timezone type.

Action method

If a policy that was defined in FUJITSU Enterprise Postgres 9.5 includes masking of a column of an unsupported data type (array type or timestamp with timezone type), perform one of the following in FUJITSU Enterprise Postgres 9.6 or later:

- Ignore the error if the masking policy contains columns of only these data types, or delete the policy with the pgx_drop_confidential_policy system administration function.
- If the masking policy target contains columns of these data types and a supported data type, delete the problematic columns with the pgx_alter_confidential_policy system function.

2.4 JDBC Driver Incompatibility

Item	Pre-migration version								
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14	14 SP1
Changing the targetServerType Value	Y	Y	Y	Y	Y	N	N	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.4.1 Changing the targetServerType Value

Incompatibility

In FUJITSU Enterprise Postgres 13, the value of targetServerType specified in the connection string was changed. Therefore, the previously used values are no longer available.

Action method

If you specified a value for targetServerType, change the value as follows:

Table 2.2 Specified values for the target server

Server Selection Order	FUJITSU Enterprise Postgres 12 or earlier	FUJITSU Enterprise Postgres 13
Primary Server	master	primary
Standby Server	slave	secondary
Prefer Standby Server	preferSlave	preferSecondary
Any	any	any

2.5 ODBC Driver Incompatibility

Item	Pre-migration version								
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14	14 SP1
Cannot specify prefer-read for target_session_attrs	Y	Y	Y	Y	Y	Y	Y	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.5.1 Cannot specify prefer-read for target_session_attrs

Incompatibility

FUJITSU Enterprise Postgres 14 users will not see the "prefer-read" radio button in the "Target_Session_Attrs" item of the data source option selection screen.

Action method

Select prefer-standby.

2.6 .NET Data Provider Incompatibility

Item	Pre-migration version								
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14	14 SP1
Changing the Specification Method and Specified Values for the Application Connection Switch Feature	Y	Y	Y	Y	Y	Y	Y	Y	Y
Changing the Behavior of the Application Connection Switch Feature	Y	Y	Y	Y	Y	Y	Y	Y	Y
Behavior Change when Multiple Host Information is Specified in the Connection String	Y	Y	Y	Y	Y	Y	Y	Y	Y
Changing the TargetServerType Value	Y	Y	Y	Y	Y	N	N	N	N
Changing the Server Explorer View in Visual Studio Integration with .NET Data Provider	Y	Y	N	N	N	N	N	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.6.1 Changing the Specification Method and Specified Values for the Application Connection Switch Feature

Incompatibility

In Fujitsu Enterprise Postgres 15, the name of the "target server" specified by the connection switching function will be changed from "TargetServerType" to "TargetSessionAttributes", and the specified value will also be changed.

Action method

If the target server is specified, change the target server and specified value as follows.

Table 2.3 How to specify the connection destination switching function of the application and the specified value

Fujitsu Enterprise Postgres version		**FUJITSU Enterprise Postgres 9.6 or earlier**	**FUJITSU Enterprise Postgres 10/11/12**	**FUJITSU Enterprise Postgres 13/14**	**Fujitsu Enterprise Postgres 15**
Keywords to specify in the connection string		target_server	TargetServerType	TargetServerType	TargetSessionAttributes
Server selection order	Primary Server	primary(*1)	master(*1)	primary(*1)	read-write(*1) primary(*2)
	Standby Server	-	slave(*2)	standby(*2)	standby read-only(*2)
	Prefer Primary Server	-	-	-	prefer-primary
	Prefer Standby Server	prefer_standby	preferSlave	preferStandby	prefer-standby
	Any	-	any	any	any

*1: A primary server whose default transaction mode is read-only are not selected.

*2: A primary server whose default transaction mode is read-only is also selected.

2.6.2 Changing the Behavior of the Application Connection Switch Feature

Incompatibility

In Fujitsu Enterprise Postgres 15, when only one Host keyword is specified in the connection string, if a target server other than Any is specified, an exception will occur.

FUJITSU Enterprise Postgres 14 SP1 earlier

For example, if the connection string specifies "Host=host1, TargetServerType=primary", it will connect to host1 if host1 is the appropriate connection target.

Fujitsu Enterprise Postgres 15

For example, if "Host=host1, TargetSessionAttributes=primary" is specified in the connection string, an exception will always occur because the target server other than Any is specified for the specification of only one host information.

Action method

None.

2.6.3 Behavior Change when Multiple Host Information is Specified in the Connection String

Incompatibility

In Fujitsu Enterprise Postgres 15, when multiple hosts are specified in the connection string, even if one of them has an empty string specified, if another connectable connection destination is specified, that connection destination will be connected. In the old version, if there was even one host with an empty string, an exception would occur at that point.

FUJITSU Enterprise Postgres 14 SP1 earlier

For example, if "Host = host1," is specified in the connection string, an exception will always occur because the second host information is an empty string.

Fujitsu Enterprise Postgres 15

For example, if the connection string specifies "Host = host1," and if host1 is reachable, it will connect to host1 without throwing an exception.

Action method

None.

2.6.4 Changing the TargetServerType Value

Incompatibility

In FUJITSU Enterprise Postgres 13, the value of TargetServerType specified in the connection string was changed. Therefore, the previously used values are no longer available.

Action method

If you specified a value for TargetServerType, change the value as follows:

Table 2.4 Specified values for the target server

Server Selection Order	FUJITSU Enterprise Postgres 9.6 or earlier	FUJITSU Enterprise Postgres 10/11/12	FUJITSU Enterprise Postgres 13
Primary Server	primary	master	primary

Server Selection Order	FUJITSU Enterprise Postgres 9.6 or earlier	FUJITSU Enterprise Postgres 10/11/12	FUJITSU Enterprise Postgres 13
Standby Server	-	slave	standby
Prefer Standby Server	prefer_standby	preferSlave	preferStandby
Any	-	any	any

2.6.5 Changing the Server Explorer View in Visual Studio Integration with .NET Data Provider

Incompatibility

FUJITSU Enterprise Postgres 10 or later, the following changes occur when you view database resources in Server Explorer.

- "Schema Name.Table Name" display under the Tables folder.
- The Indexes folder is not displayed.
- The Triggers folder does not displayed.
- The Sequences folder is not displayed.
- The Procedures folder does not displayed.

Action method

None.

2.7 C Library (libpq) Migration Incompatibility

Item	Pre-migration version								
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14	14 SP1
Changing when "prefer-read" is Specified for the target_session_attrs Parameter	Y	Y	Y	Y	Y	Y	Y	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.7.1 Changing when "prefer-read" is Specified for the target_session_attrs Parameter

Incompatibility

In FUJITSU Enterprise Postgres 14 changes the attach server priority if any of the following servers are specified simultaneously with "prefer-read" as the target_session_attrs parameter:

- Primary server (default_transaction_read_only = ON)
- Standby server

FUJITSU Enterprise Postgres 13 SP1 or earlier

The primary server (default_transaction_read_only = ON) and standby servers have the same priority.

FUJITSU Enterprise Postgres 14 or later

Standby servers connect in preference to primary servers (default_transaction_read_only = ON).

Action method

None.

2.8 oracle_fdw Incompatibility

Item	Pre-migration version								
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14	14 SP1
Changing the Oracle Client Version	N	N	Y	Y	N	N	N	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.8.1 Changing the Oracle Client Version

Incompatibility

FUJITSU Enterprise Postgres 12 change the version of the Oracle client used to build oracle_fdw to 11.2.

Action method

Use Oracle client version 11.2 or later.

Also,if a file named libclntsh.so.11.1 does not exist in OCI library, create a symbolic link named libclntsh.so.11.1.

2.9 pgaudit Incompatibility

Item	Pre-migration version								
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14	14 SP1
Changing to Output Extra NEW and OLD Values in the Audit Log when the Trigger Function Executes	N	N	Y	Y	Y	N	N	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.9.1 Changing to Output Extra NEW and OLD Values in the Audit Log when the Trigger Function Executes

Incompatibility

In FUJITSU Enterprise Postgres 13, the trigger function additionally outputs NEW and OLD values to the audit log when the pgaudit.log_parameter is set to on.

FUJITSU Enterprise Postgres 12 or earlier

The following (18) does not output the values of NEW, OLD.

[Example]

```
AUDIT: SESSION,WRITE,2020-09-03 07:07:39 UTC,
        (1)    (2)   (3)
```

```
[local],9775,psql,k5user,postgres,3/536, 1, 2, INSERT, ,
(4)     (5)  (6)  (7)    (8)      (9)   (10)(11)(12)(13)
TABLE,public.trig_audit, ,
(14)  (15)    (16)
 "INSERT INTO trig_audit SELECT 'U', now(), user, OLD.*, NEW.*",
(17)
trig_audit AFTER ROW UPDATE 92027 trig_test trig_test public 0  f aaaa
(18)
```

FUJITSU Enterprise Postgres 13 or later

NEW, OLD values are output.

[Example]

```
AUDIT: SESSION,WRITE,2020-09-03 07:07:39 UTC,
[local],9775,psql,k5user,postgres,3/536,1,2,INSERT,,TABLE,public.
trig_audit,,
"INSERT INTO trig_audit SELECT 'U', now(), user, OLD.*, NEW.*",
(bbb) (aaa) trig_audit AFTER ROW UPDATE 92027 trig_test trig_test
public 0  f aaaa
```

Action method

If you are using an application that works by monitoring the string that the trigger function output to the audit log, modify the application to work with the NEW and OLD values.

2.10 WebAdmin Incompatibility

Item	Pre-migration version								
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14	14 SP1
Cannot specify prefer-read for target_session_attrs	Y	Y	Y	Y	Y	Y	Y	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.10.1 Cannot specify prefer-read for target_session_attrs

Incompatibility

FUJITSU Enterprise Postgres 14 will no longer allow prefer-read to target_session_attrs as a connection method to an upstream server that is specified when creating an instance of a standby server.

Action method

Specify prefer-standby.

2.11 Connection Manager Incompatibility

Item	Pre-migration version				
	12	13	12 SP1/13 SP1	14	14 SP1
Behavior change when "read-write" is specified for the target_session_attrs parameter	Y	Y	Y	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.11.1 Behavior change when "read-write" is specified for the target_session_attrs parameter

Incompatibility

FUJITSU Enterprise Postgres 13 SP1 or earlier

May be connected to primary server (default_transaction_read_only = ON).

FUJITSU Enterprise Postgres 14 or later

It is not connected to the primary server (default_transaction_read_only = ON).

Action method

For FUJITSU Enterprise Postgres 13 SP1 and earlier, specify "primary" for the target_session_attrs parameter.

Chapter 3 Program Updates

For program fix information, see "Program Updates".

Index

[C]

Compatibility Information.....5

[F]

Features Added in 15.....1

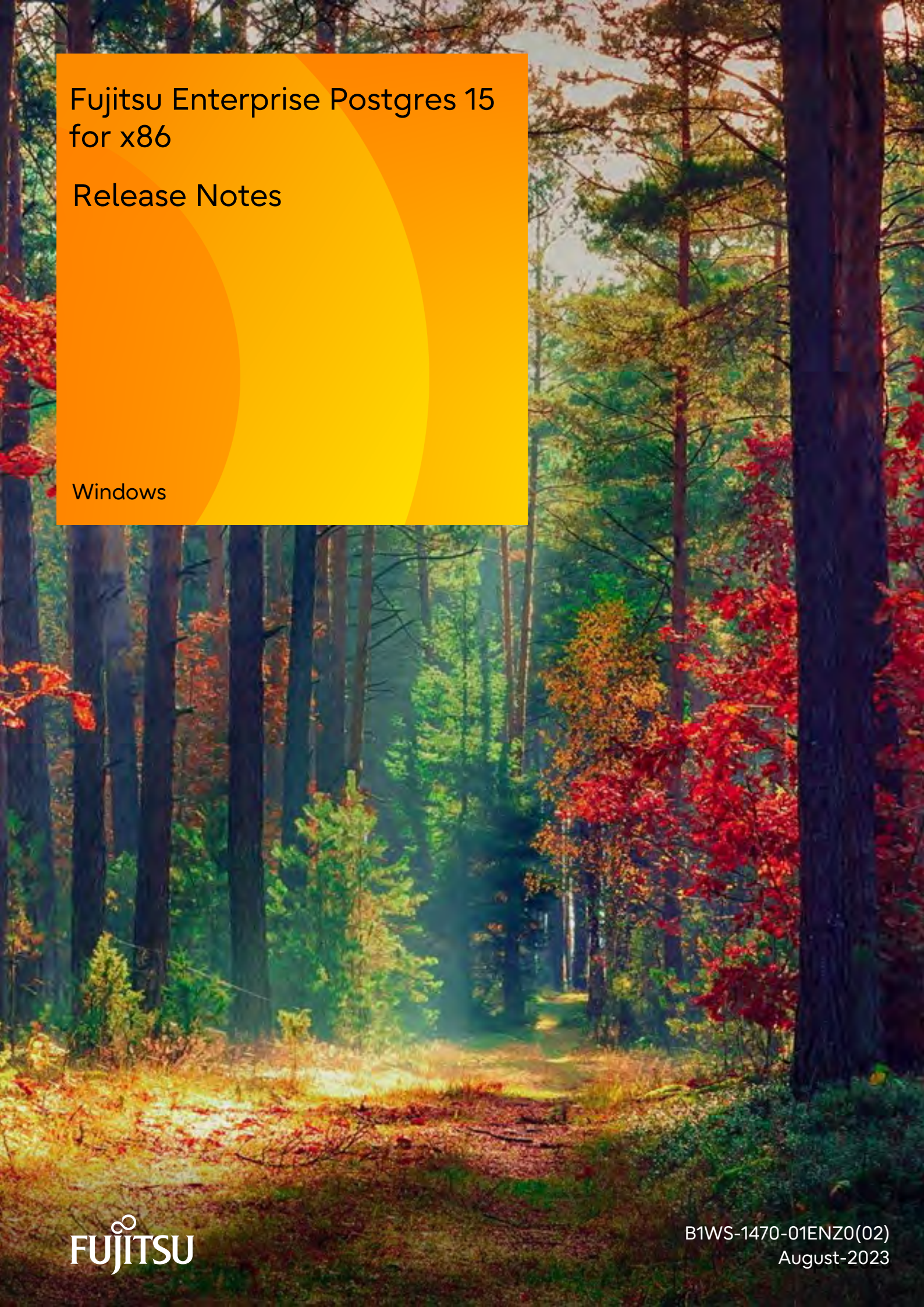
[P]

Program Updates.....33

Fujitsu Enterprise Postgres 15 for x86

Release Notes

Windows



FUJITSU

B1WS-1470-01ENZ0(02)
August-2023

Preface

Purpose of this document

This document provides release information for Fujitsu Enterprise Postgres.

Structure of this document

This document is structured as follows:

Chapter 1 New Features and Improvements

Explains the new features and improvements in this version.

Chapter 2 Compatibility Information

Provides information regarding compatibility.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

```
Edition 1.2: August 2023
Edition 1.1: June 2023
Edition 1.0: April 2023
```

Copyright

Copyright 2015-2023 Fujitsu Limited

Revision History

Revision	Location	Version
Added an incompatible article about migrating applications.	2.2.1 Changing the OID of the Data Type (NCHAR type) that Handles National Characters	Edition 1.2
Feature names have been improved. Before modification : Confidential management support After modification : Confidentiality management	Table 1.1 New features and improvements	Edition 1.1
	1.1.1 Security	
	1.1.1.1 Confidentiality Management	

Contents

Chapter 1 New Features and Improvements....1
1.1 Features Added in 15....1
1.1.1 Security....1
1.1.1.1 Confidentiality Management....1
1.1.2 Auditing....1
1.1.2.1 pgaudit Improvements....1
1.1.3 OSS....2
1.1.3.1 PostgreSQL Rebase....2
1.1.3.2 OSS Updates Provided....2
1.1.4 Platform Enhancement....2
1.1.4.1 Additional Operating System Support for Client Feature....2
1.1.4.2 Additional Operating System Support for Server Assistant Feature....2
1.1.5 Application Development....2
1.1.5.1 Addition of JDKs and JREs on which the JDBC Driver can Work....3
1.1.6 Miscellaneous (LD_LIBRARY_PATH Improvements)....3
1.1.6.1 Additonal the DT_RUNPATH Attribute....3
1.1.6.2 About Building Applications and Extensions....3

Chapter 2 Compatibility Information....4
2.1 Installation/Setup Incompatibility....4
2.1.1 Removing Operating System Support for Client Feature....4
2.1.2 Removing Operating System Support for Server Feature....4
2.1.3 Removing Operating System Support for Client Feature....5
2.1.4 Removing Operating System Support for Client Feature....5
2.1.5 Removing Operating System Support for Server Feature....5
2.1.6 Changing the Way OSS is Set Up....5
2.1.7 Changing Core and Log File Paths when Instance is Created with WebAdmin....6
2.1.8 Renaming WebAdmin Services....6
2.1.9 Changing the Features Targeted for Installation in a 64-bit Environment....6
2.1.10 Changing the Access Permissions of the Windows Client Installation Folder....7
2.1.11 Changing the WebAdmin Installation Method....7
2.2 Application Migration Incompatibility....7
2.2.1 Changing the OID of the Data Type (NCHAR type) that Handles National Characters....8
2.2.2 Changed the Specification Method of the Application Connection Switch Feature....8
2.2.3 Changes in ecpg/ecobpg Handling of Zero-Length Unicode Identifiers....9
2.2.4 Changing the display result when data masking is applied to NaN, infinity, -infinity....9
2.2.5 Changing the Valid Range of Identifiers Defined by the DECLARE STATEMENT statement....9
2.2.6 Changing Precompile Results....10
2.2.7 Changing the Trigger Replacement Process....10
2.2.8 Change the "elif" Statement Behavior in ecobpg....10
2.2.9 Changed to Error when Running an Operator or Function that Returns non Data Types for Masking Type....11
2.2.10 Removing UTL_FILE for Oracle Database Compatibility Features....11
2.2.11 Changing the Method of Specifying the Application Connection Switch Feature....12
2.2.12 PostgreSQL Compatibility of Embedded SQL Applications in C and COBOL....12
2.2.13 Changing Vertical Clustered Index (VCI)....12
2.2.14 Changing how to Use the Features Compatible with Oracle Databases....13
2.3 Operation Migration Incompatibility....13
2.3.1 Abolition of Message Numbers....14
2.3.2 Rename column "master_pid" in pgx_loader_state to "leader_pid"....15
2.3.3 Changing the Value of the Category Column in the pg_settings view....15
2.3.4 Changing pgx_stat_lwlock of the Statistics View....15
2.3.5 Changing the Behavior of pgx_rcvall....16
2.3.6 Mirroring Controller no Longer Retries to Monitor Database Processes when they are Detected as Down....17
2.3.7 Changing the Name and Parameter Name of the Mirroring Controller Post-Promote Command....17
2.3.8 Changing Mirroring Controller User Command Input Values....17

2.3.9 Changing the Maximum Number of Connections per Server ... 18
2.3.10 Changing the Encryption Settings Using the ALTER TABLESPACE Statement ... 18
2.3.11 Changing the Default Configuration of the Cluster System Using Database Multiplexing ... 18
2.3.12 Changing the Default Operation when mc_ctl Command Options are Omitted ... 18
2.3.13 Changing the Connection Settings when Mirroring Controller Connects to an Instance ... 19
2.3.14 Changing the Status Display of the Mirroring Controller Server ... 19
2.3.15 Changing the Operation when the synchronous_standby_names Parameter is Changed during Database Multiplexing Operation ... 20
2.3.16 Changing Masking Policy Definition for Unsupported Data Types ... 20
2.4 JDBC Driver Incompatibility ... 21
2.4.1 Changing the targetServerType Value ... 21
2.5 ODBC Driver Incompatibility ... 21
2.5.1 Cannot specify prefer-read for target_session_attrs ... 22
2.6 .NET Data Provider Incompatibility ... 22
2.6.1 Changing the Specification Method and Specified Values for the Application Connection Switch Feature ... 22
2.6.2 Changing the Behavior of the Application Connection Switch Feature ... 23
2.6.3 Behavior Change when Multiple Host Information is Specified in the Connection String ... 23
2.6.4 Changing the TargetServerType Value ... 24
2.6.5 Changing the Server Explorer View in Visual Studio Integration with .NET Data Provider ... 24
2.7 C Library (libpq) Migration Incompatibility ... 24
2.7.1 Changing when "prefer-read" is Specified for the target_session_attrs Parameter ... 24
2.8 pgaudit Incompatibility ... 25
2.8.1 Changing to Output Extra NEW and OLD Values in the Audit Log when the Trigger Function Executes ... 25
2.9 WebAdmin Incompatibility ... 26
2.9.1 Cannot specify prefer-read for target_session_attrs ... 26

Chapter 3 Program Updates ... 27

Index ... 28

Chapter 1 New Features and Improvements

This chapter explains Fujitsu Enterprise Postgres new features and improvements added in this version.

Table 1.1 New features and improvements

Version and level	Classification	Feature	Provided in AE	Provided in SE
15	Security	Confidentiality Management	Y	N
	Auditing	pgaudit Improvements	Y	N
	OSS	PostgreSQL Rebase	Y	Y
		OSS updates provided	Y	Y
	Platform enhancement	Additional Operating System Support for Client Feature	Y	Y
		Additional Operating System Support for Server Assistant Feature	Y	Y
	Application Development	Add JDKs and JREs that can run JDBC drivers	Y	Y

1.1 Features Added in 15

This section explains new features and improvements in Fujitsu Enterprise Postgres 15.

1.1.1 Security

This section explains the new features and improvements related to security:

- Confidentiality Management

1.1.1.1 Confidentiality Management

The confidentiality management function is provided to assist in the design and operation of access control for databases. This reduces the man-hours required to manage complicated access control.

See

Refer to the Security Operation Guide for more information.

1.1.2 Auditing

This section describes new features and improvements related to auditing.

- pgaudit Improvements

1.1.2.1 pgaudit Improvements

The audit_log_disconnections parameter has been added.

See

Refer to Security Operations Guide for more information,

1.1.3 OSS

This section explains the new feature related to OSS:

- PostgreSQL Rebase
- OSS Updates Provided

1.1.3.1 PostgreSQL Rebase

The PostgreSQL version that Fujitsu Enterprise Postgres is based on is 15.0.

See

Refer to "PostgreSQL Version Used for Fujitsu Enterprise Postgres" in the Installation and Setup Guide for Server for more information.

1.1.3.2 OSS Updates Provided

The OSS provided by Fujitsu Enterprise Postgres has been updated.

See

Refer to "OSS Supported by Fujitsu Enterprise Postgres" in the General Description for details.

1.1.4 Platform Enhancement

This section explains the new features related to platform enhancement:

- Additional operating system support for client
- Additional operating system support for server assistant

1.1.4.1 Additional Operating System Support for Client Feature

The following additional operating system is supported:

- RHEL9

See

Refer to "Required Operating System" in the Installation and Setup Guide for Server for more information,

1.1.4.2 Additional Operating System Support for Server Assistant Feature

The following additional operating system is supported:

- RHEL9

See

Refer to "Required Operating System" in the Installation and Setup Guide for Server Assistant for more information,

1.1.5 Application Development

This section describes new features related to application development.

- Addition of JDKs and JREs on which the JDBC Driver can Work

1.1.5.1 Addition of JDKs and JREs on which the JDBC Driver can Work

The following JDKs and JREs have been added to the list of supported JDKs and JREs.

- JDK 17 and JRE 17

See

Refer to "Related Software" in the Installation Guide for Client for details,

1.1.6 Miscellaneous (LD_LIBRARY_PATH Improvements)

Describes the LD_LIBRARY_PATH improvements.

- Additonal the DT_RUNPATH Attribute
- About Building Applications and Extensions

1.1.6.1 Additonal the DT_RUNPATH Attribute

Add DT_RUNPATH attribute to Fujitsu Enterprise Postgres programs.

This allows Fujitsu Enterprise Postgres programs to run without setting the environment variable LD_LIBRARY_PATH to the path to the installation directory/lib for Fujitsu Enterprise Postgres features.

1.1.6.2 About Building Applications and Extensions

If you are building an application that makes use of the shared libraries shipped with Fujitsu Enterprise Postgres, or an extension module for Fujitsu Enterprise Postgres, ensure that the DT_RUNPATH attribute of the program is set to the path to the directory that contains those libraries.

This allows programs to run without setting the environment variable LD_LIBRARY_PATH to the path to the Fujitsu Enterprise Postgres installation directory/lib.

See

Refer to about setting the DT_RUNPATH attribute, see:

- "How to Build and Run an Application that Uses Shared Libraries" in the Application Development Guide
- "Build with PGXS" and "Build without PGXS" in "Setting Up and Removing OSS" in the Installation and Setup Guide for Server

You can also continue to set the environment variable LD_LIBRARY_PATH to the path to the Fujitsu Enterprise Postgres installation directory/lib, and run these programs without setting the DT_RUNPATH attribute.

See

For notes on setting the environment variable LD_LIBRARY_PATH, refer to "When DT_RUNPATH cannot be set" in "How to Build and Run an Application that Uses Shared Libraries" in the Application Development Guide.

Chapter 2 Compatibility Information

This chapter explains incompatible items and actions required when migrating from an earlier version to Fujitsu Enterprise Postgres 15. Check compatibility before migrating and take the appropriate action.

2.1 Installation/Setup Incompatibility

Item	Pre-migration version							
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14
Removing Operating System Support for Client Feature	Y	Y	Y	Y	Y	Y	N	N
Removing Operating System Support for Server Feature	Y	Y	Y	Y	Y	Y	N	N
Removing Operating System Support for Client Feature	N	N	Y	Y	Y	N	N	N
Removing Operating System Support for Client Feature	Y	Y	Y	Y	N	N	N	N
Removing Operating System Support for Server Feature	Y	Y	Y	Y	N	N	N	N
Changing the way OSS is set up	Y	Y	Y	Y	N	N	N	N
Changing Core and Log File Paths when Instance is Created with WebAdmin	Y	Y	Y	Y	N	N	N	N
Renaming WebAdmin Services	Y	Y	Y	Y	N	N	N	N
Changing the features targeted for installation in a 64-bit environment	Y	Y	N	N	N	N	N	N
Changing the access permissions of the Windows client installation folder	Y	Y	N	N	N	N	N	N
Changing the WebAdmin installation method	Y	N	N	N	N	N	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.1.1 Removing Operating System Support for Client Feature

Incompatibility

In FUJITSU Enterprise Postgres 13 SP1 or later, the following operating systems have been removed.

- Windows Server(R) 2012
- Windows Server(R) 2012 R2

Action method

None.

2.1.2 Removing Operating System Support for Server Feature

Incompatibility

In FUJITSU Enterprise Postgres 13 SP1 or later, the following operating systems have been removed.

- Windows Server(R) 2012
- Windows Server(R) 2012 R2

Action method

None.

2.1.3 Removing Operating System Support for Client Feature

Incompatibility

In FUJITSU Enterprise Postgres 13 or later, the following operating systems have been removed.

- SLES 12 SP4 or earlier

Action method

None.

2.1.4 Removing Operating System Support for Client Feature

Incompatibility

In FUJITSU Enterprise Postgres 12 or later, the following operating systems have been removed.

- Windows(R) 7
- Windows Server(R) 2008
- RHEL6

Also, the 32 bit Linux client can no longer be installed because RHEL 6 has been removed from the operating system.

Action method

None.

2.1.5 Removing Operating System Support for Server Feature

Incompatibility

In FUJITSU Enterprise Postgres 12 or later, the following operating systems have been removed.

- Windows Server(R) 2008

Action method

None.

2.1.6 Changing the Way OSS is Set Up

Incompatibility

FUJITSU Enterprise Postgres 12 or later do not place OSS extension modules in the executable directory. The OSS extension modules must be placed in the executable directory when you set up OSS.

Refer to "Setting Up and Removing OSS" in the Installation and Setup Guide for Server for details.

Action method

None.

2.1.7 Changing Core and Log File Paths when Instance is Created with WebAdmin

Incompatibility

In FUJITSU Enterprise Postgres 12 or later, change the core and log file paths when creating an instance in WebAdmin.

FUJITSU Enterprise Postgres 11 or earlier

Log File Path: *userProfileFolder**localSettingsFolder*\Fujitsu\fsep_*version**instanceNamePortNumber*\log

Core File Path: *userProfileFolder**localSettingsFolder*\Fujitsu\fsep_*version**instanceNamePortNumber*\core

version: product version_edition_architecture

［Example］

Log File Path: C:\Users\naomi\AppData\Local\Fujitsu\fsep_110_AE_64\myinst27599\log

Core File Path: C:\Users\naomi\AppData\Local\Fujitsu\fsep_110_AE_64\myinst27599\core

FUJITSU Enterprise Postgres 12

*userProfileFolder**localSettingsFolder*\Fujitsu\fsep_*version**instanceNamePortNumber*\log

Core File Path: *userProfileFolder**localSettingsFolder*\Fujitsu\fsep_*version**instanceNamePortNumber*\core

version: product version_WA_architecture

［Example］

Log File Path: C:\Users\naomi\AppData\Local\Fujitsu\fsep_120_WA_64\myinst27599\log

Core File Path: C:\Users\naomi\AppData\Local\Fujitsu\fsep_120_WA_64\myinst27599\core

Action method

None.

2.1.8 Renaming WebAdmin Services

Incompatibility

In FUJITSU Enterprise Postgres 12 or later, change the service name registered when you set up WebAdmin.

FUJITSU Enterprise Postgres 11 or earlier

fsep_*x*SP*z*_*edition*_64_WebAdmin_Port1

fsep_*x*SP*z*_*edition*_64_WebAdmin_Port2

FUJITSU Enterprise Postgres 12 or later

fsep_xSPz_WA_64_WebAdmin_Port1

fsep_xSPz_WA_64_WebAdmin_Port2

Action method

None.

2.1.9 Changing the Features Targeted for Installation in a 64-bit Environment

Incompatibility

32 bit Linux client can no longer be installed on RHEL7 in FUJITSU Enterprise Postgres 10 or later.

Note: 32 bit Linux client can install on RHEL6 in FUJITSU Enterprise Postgres 11 or earlier.

Action method

None.

2.1.10 Changing the Access Permissions of the Windows Client Installation Folder

Incompatibility

When changed for install folder the Windows client from default, the access permissions of the installed files and folder changes to read and execute.

The above changes apply to non-administrators.

This incompatibility occurs with all supported Windows client operating systems. Refer to "Required Operating System" in the Installation and Setup Guide for Client for information on the operating environment.

Action method

This change was made simply to enhance security, therefore, no action is normally required.

However, if existing access permissions are required, this change can be reverted by running *installDir*\setup\revert_cacls.bat as an administrator after installation.

2.1.11 Changing the WebAdmin Installation Method

Incompatibility

In FUJITSU Enterprise Postgres 9.6 or later, WebAdmin is not installed automatically during server installation. Therefore, install WebAdmin separately.

Action method

None.

2.2 Application Migration Incompatibility

Item	Pre-migration version							
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14
Changing the OID of the Data Type (NCHAR type) that Handles National Characters	Y	Y	Y	Y	Y	Y	Y	Y
Changed the Specification Method of the Connection Destination Switching Function of the Application	Y	Y	Y	Y	Y	Y	Y	Y
Changes in ecpg/ecobpg Handling of Zero-Length Unicode Identifiers	Y	Y	Y	Y	Y	Y	Y	Y
Changing the display result when data masking is applied to NaN, infinity, -infinity	Y	Y	Y	Y	Y	Y	Y	N

Item	Pre-migration version							
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14
Changing the Valid Range of Identifiers Defined by the DECLARE STATEMENT statement	Y	Y	Y	Y	Y	Y	Y	N
Changing Precompile Results	Y	Y	Y	Y	Y	Y	Y	N
Changing the Trigger Replacement Process	Y	Y	Y	Y	Y	Y	Y	N
Change the "elif" Statement Behavior in ecobpg	Y	Y	Y	Y	Y	N	N	N
Changed to Error when Running an Operator or Function that Returns non Data Types for Masking Type	Y	Y	Y	Y	N	N	N	N
Removing UTL_FILE for Oracle Database Compatibility Features	Y	Y	Y	Y	N	N	N	N
Changing the method of specifying the application connection switch feature	Y	Y	N	N	N	N	N	N
PostgreSQL compatibility of embedded SQL applications in C and COBOL	Y	Y	N	N	N	N	N	N
Changing Vertical Clustered Index (VCI)	Y	Y	N	N	N	N	N	N
Changing how to use the features compatible with Oracle databases	Y	N	N	N	N	N	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.2.1 Changing the OID of the Data Type (NCHAR type) that Handles National Characters

Incompatible

In Fujitsu Enterprise Postgres 15, OIDs for national character data types (NCHAR types) have changed.

Action method

If you are using a national character data type (NCHAR type), recompile the application and run it with Fujistu Enterprise Postgres 15 or later clients.

2.2.2 Changed the Specification Method of the Application Connection Switch Feature

Incompatible

In Fujitsu Enterprise Postgres 15, the name of the "target server" specified by the application connection switch feature will be changed.

Action method

For applications that specify a target server as the application connection switch feature, change the target server according to the table below, recompile, and then execute using a client of Fujistu Enterprise Postgres 15 or later.

Table 2.1 Specified name of the target server.

Target of use	Before modification	After modification
Leverage .NET Data Provider	TargetServerType	TargetSessionAttributes

2.2.3 Changes in ecpg/ecobpg Handling of Zero-Length Unicode Identifiers

Incompatible

In Fujitsu Enterprise Postgres 15, if the SQL statement contains a zero-length Unicode identifier, Change precompilation to fail.

Action method

None.

2.2.4 Changing the display result when data masking is applied to NaN, infinity, -infinity

Incompatibility

In FUJITSU Enterprise Postgres 14, the display result when data masking is applied to NaN, infinity, and -infinity will be changed.

FUJITSU Enterprise Postgres 13 SP1 or earlier

If the float type NaN, infinity, and -infinity are partially masking with (9,1,2), the following will be displayed.

```
NaN : 99
Infinity : 99
-Infinity : 99
```

FUJITSU Enterprise Postgres 14 or later

If the float type NaN, infinity, and -infinity are partially masking with (9,1,2), the following will be displayed.

```
NaN : NaN
Infinity : Infinity
-Infinity : -Infinity
```

Action method

If the application is analyzing the SQL masking output result, please consider the non-numeric output result and correct it.

2.2.5 Changing the Valid Range of Identifiers Defined by the DECLARE STATEMENT statement

Incompatibility

In FUJITSU Enterprise Postgres 14 will change the valid range of identifiers defined by a DECLARE STATEMENT statement in ecpg/ecobpg.

FUJITSU Enterprise Postgres 13 SP1 or earlier

The valid range is now per process.

FUJITSU Enterprise Postgres 14 or later

The valid range is now per file.

Action method

None.

2.2.6 Changing Precompile Results

Incompatibility

In FUJITSU Enterprise Postgres 14 removed the ECPGdeclare/ECPGopen function. Therefore, results precompiled from earlier versions of FUJITSU Enterprise Postgres will not be available in FUJITSU Enterprise Postgres 14.

Action method

Rebuild the application.

2.2.7 Changing the Trigger Replacement Process

Incompatibility

In FUJITSU Enterprise Postgres 14 will change restricted triggers to not be supported by replace operations (OR REPLACE).

FUJITSU Enterprise Postgres 13 SP1 or earlier

You can replace a constraint trigger.

[Example]

```
=# CREATE OR REPLACE CONSTRAINT TRIGGER my_constraint_trigger AFTER DELETE ON my_table
-# FOR EACH ROW
-# EXECUTE PROCEDURE funcA();
CREATE TRIGGER
```

FUJITSU Enterprise Postgres 14 or later

It does not support replacing a constraint trigger.

[Example]

```
=# CREATE OR REPLACE CONSTRAINT TRIGGER my_constraint_trigger AFTER DELETE ON my_table
-# FOR EACH ROW
-# EXECUTE PROCEDURE funcA();
ERROR:  CREATE OR REPLACE CONSTRAINT TRIGGER is not supported
```

Action method

None.

2.2.8 Change the "elif" Statement Behavior in ecobpg

Incompatibility

The "elif" statement now works correctly with FUJITSU Enterprise Postgres 13.

FUJITSU Enterprise Postgres 12 or earlier

The decision of the "elif" statement is made whether or not the "ifdef/ifndef" condition is met.

[Example]

```
000000     EXEC SQL ifdef DEF1 END-EXEC.
000000     Operation (1).
000000     EXEC SQL elif DEF2 END-EXEC.
000000     Operation (2).
000000     EXEC SQL else END-EXEC.
000000     Operation (3).
```

If "DEF1" and "DEF2" are true, both "Operation (1)" and "Operation (2)" are executed.

FUJITSU Enterprise Postgres 13 or later

If the condition "ifdef/ifndef" is met, the judgment of the "elif" statement is not enforced.

[Example]

```
000000     EXEC SQL ifdef DEF1 END-EXEC.
000000     Operation (1).
000000     EXEC SQL elif DEF2 END-EXEC.
000000     Operation (2).
000000     EXEC SQL else END-EXEC.
000000     Operation (3).
```

If "DEF1" and "DEF2" are true, only "Operation (1)" is executed.

Action method

Check your existing application and fix the elif statement to work correctly.

2.2.9 Changed to Error when Running an Operator or Function that Returns non Data Types for Masking Type

Incompatibility

In FUJITSU Enterprise Postgres 12, changed to error when running an operator or function that returns non Data Types for Masking type.

FUJITSU Enterprise Postgres 11 or earlier

In the following cases, the operator or function in 3) may be executed without masking.

This is an incorrect result because the data containing the masking column is performed without masking.

1) Create a Masking policy. and

2) Execute a SELECT statement. and

3) Execute an operator or function in the SELECT statement of 2). and

4) The operator or function argument of 3) includes a subquery. and

5) The operator or function argument of 3) contains the column to be protected created in 1). and

6) The operator or function of 3) returns a type not listed in "Data Types for Masking" in "Operation Guide".

FUJITSU Enterprise Postgres 12 or later

Operator or function results of 3) in the following error.

```
ERROR:  The output data type is incompatible with the confidential policy.
HINT:  Data type of the result value(s) produced by expression/function using confidential columns
is not supported by Data masking module. Consider removing confidential columns from such
expressions/functions.
```

This is correct because the operator or function returns a data type not listed in "Data Types for Masking" in "Operation Guide".

Action method

Do one of the following to ensure that the results are identical to those of FUJITSU Enterprise Postgres 11 and earlier:

1) Modify the Masking policy to prevent masking from being performed for the user executing the SQL.

2) Modify SQL to not use operators or functions that return types not listed in "Data Types for Masking" in "Operation Guide".

2.2.10 Removing UTL_FILE for Oracle Database Compatibility Features

Incompatibility

In FUJITSU Enterprise Postgres 12, remove the Oracle database compatibility UTL_FILE.

Action method

None.

2.2.11 Changing the Method of Specifying the Application Connection Switch Feature

Incompatibility

The target server specified using the application connection switch feature is changed to the same name as that of PostgreSQL.

Action method

Use the FUJITSU Enterprise Postgres 10 or later client to execute applications that have the target server specified as the application connection switch feature after changing the target server and specified value as listed below and recompiling.

Specified name of the target server

Usage target	Before correction	After correction
JDBC driver	No change	
ODBC driver	TargetServer	target_session_attrs
.NET Data Provider	target_server	TargetServerType
Connection service file	target_server	target_session_attrs
Library (libpq) for C	target_server Environment variable (PGXTARGETSERVER)	target_session_attrs Environment variable (PGTARGETSESSIONATTRS)
Embedded SQL	target_server	target_session_attrs
psql	Environment variable (PGXTARGETSERVER)	Environment variable (PGTARGETSESSIONATTRS)

Specified value for the target server

Server selection order	JDBC driver	.NET Data Provider		Other driver	
		Before correction	After correction	Before correction	After correction
Primary server	No change	primary	master	primary	read-write
Standby server	No change	-	slave	-	-
Priority given to a standby server	No change	prefer_standby	preferSlave	prefer_standby	prefer-read
Any	No change	-	any	-	any

2.2.12 PostgreSQL Compatibility of Embedded SQL Applications in C and COBOL

Incompatibility

Embedded SQL applications in C and COBOL that were compiled with FUJITSU Enterprise Postgres 9.6 or earlier cannot be used in the FUJITSU Enterprise Postgres 10 client.

Action method

Use the FUJITSU Enterprise Postgres 10 or later client to execute the applications only after recompiling them with the FUJITSU Enterprise Postgres 10 or later client.

2.2.13 Changing Vertical Clustered Index (VCI)

Incompatibility

The functions below have been added to the functions for which VCI is not used.

Type	Function
Date/time functions	age(timestamp) current_date current_timestamp current_time localtime localtimestamp
Session information functions	current_user current_role

Action method

To use VCI, specify the second argument when using the age function, and specify other functions as subqueries.

[Example]

Before: select age(column A), current_date from table

After: select age(column A, **now()**), **(select current_date)** from table

2.2.14 Changing how to Use the Features Compatible with Oracle Databases

Incompatibility

In FUJITSU Enterprise Postgres 9.6 or later, to use the features compatible with Oracle databases, create a new instance and execute the following command for the "postgres" and "template1" databases:

CREATE EXTENSION oracle_compatible

Action method

None.

2.3 Operation Migration Incompatibility

Item	Pre-migration version							
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14
Abolition of Message Numbers	Y	Y	Y	Y	Y	Y	Y	Y
Rename column "master_pid" in pgx_loader_state to "leader_pid"	N	N	Y	Y	Y	Y	Y	N
Changing the Value of the Category Column in the pg_settings view	N	Y	Y	Y	Y	N	N	N
Changing pgx_stat_lwlock of the Statistics View	N	Y	Y	Y	Y	N	N	N
Changing the Behavior of pgx_rcvall	Y	Y	Y	Y	Y	N	N	N

Item	Pre-migration version							
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14
Mirroring Controller no longer retries to monitor database processes when they are detected as down	Y	N	Y	Y	N	N	N	N
Changing the Name and Parameter Name of the Mirroring Controller Post-Promote Command	N	N	Y	Y	N	N	N	N
Changing Mirroring Controller User Command Input Values	N	N	Y	Y	N	N	N	N
Changing the maximum number of connections per server	Y	Y	N	N	N	N	N	N
Changing the encryption settings using the ALTER TABLESPACE statement	Y	Y	N	N	N	N	N	N
Changing the default configuration of the cluster system using database multiplexing	Y	N	N	N	N	N	N	N
Changing the default operation when mc_ctl command options are omitted	Y	N	N	N	N	N	N	N
Changing the connection settings when Mirroring Controller connects to an instance	Y	N	N	N	N	N	N	N
Changing the status display of the Mirroring Controller server	Y	N	N	N	N	N	N	N
Changing the operation when the synchronous_standby_names parameter is changed during database multiplexing operation	Y	N	N	N	N	N	N	N
Changing masking policy definition for unsupported data types	Y	N	N	N	N	N	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.3.1 Abolition of Message Numbers

Incompatibility

In Fujitsu Enterprise Postgres 15, the message number output at the end of the message is abolished.

Message numbers are output for messages output by Mirroring Controller.

For FUJITSU Enterprise Postgres 14 or earlier

The message number was printed at the end of the message.

[example]

```
3D000: 2017-07-10 19:41:05 JST[13899]: [1-1] user=fepuser,db=fep,remote=127.0.0.1(51902)
app=[unknown] FATAL: database "fep" does not exist (10571)
```

For Fujitsu Enterprise Postgres 15

No message number is output at the end of the message.

[example]

```
3D000: 2023-04-10 19:41:05 JST [13899]: [1-1] user = fepuser,db = fep,remote = 127.0.0.1(51902)
app = [unknown] FATAL: database "fep" does not exist
```

Action method

None.

2.3.2 Rename column "master_pid" in pgx_loader_state to "leader_pid"

Incompatibility

In FUJITSU Enterprise Postgres 14 renames column "master_pid" to "leader_pid" in the pgx_loade_state table.

Action method

None.

2.3.3 Changing the Value of the Category Column in the pg_settings view

Incompatibility

For FUJITSU Enterprise Postgres 13, change the value of the category column in the pg_settings view.

FUJITSU Enterprise Postgres 12 or earlier

Original value	wrong value
Preset Options	Fujitsu Enterprise Postgres Parameters
Customized Options	Preset Options
Developer Options	Customized Options

This is an incorrect result because it is different from the original value.

FUJITSU Enterprise Postgres 13 or later

The correct value is the category column in the pg_settings view.

Action method

Replaces the category column in the pg_settings view with the values before migration, so that the results are the same as before migration.

2.3.4 Changing pgx_stat_lwlock of the Statistics View

Incompatibility

In FUJITSU Enterprise Postgres 13, change the wait event name displayed in the lwlock_name column of the statistics view pgx_stat_lwlock.

Wait Event Name

FUJITSU Enterprise Postgres 12 or earlier	FUJITSU Enterprise Postgres 13
clog (*1)	XactBuffer
commit_timestamp (*1)	CommitTSBuffer
subtrans (*1)	SubtransBuffer
multixact_offset (*1)	MultiXactOffsetBuffer
multixact_member (*1)	MultiXactMemberBuffer
async (*1)	NotifyBuffer

FUJITSU Enterprise Postgres 12 or earlier	FUJITSU Enterprise Postgres 13
oldserxid (*1)	SerialBuffer
wal_insert (*1)	WALInsert
buffer_content (*1)	BufferContent
buffer_io (*1)	BufferIO
replication_origin (*1)	ReplicationOriginState
replication_slot_io (*1)	ReplicationSlotIO
proc (*1)	LockFastPath
buffer_mapping (*1)	BufferMapping
lock_manager (*1)	LockManager
predicate_lock_manager (*1)	PredicateLockManager
parallel_hash_join (*3)	ParallelHashJoin
parallel_query_dsa (*2)	ParallelQueryDSA
session_dsa (*3)	PerSessionDSA
session_record_table (*3)	PerSessionRecordType
session_typmod_table (*3)	PerSessionRecordTypmod
shared_tuplestore (*3)	SharedTupleStore
tbm (*2)	SharedTidBitmap
parallel_append (*3)	ParallelAppend
serializable_xact (*4)	PerXactPredicateList
shared_mcxt (*4)	SharedMcxt
meta_cache_map (*4)	MetaCacheMap
global_metacache (*4)	GlobalCatcache
cached_buf_tranche_id (*4)	CachedBufTranche

*1）Events added in FUJITSU Enterprise Postgres 9.6.

*2）Events added in FUJITSU Enterprise Postgres 10.

*3）Events added in FUJITSU Enterprise Postgres 11.

*4）Events added in FUJITSU Enterprise Postgres 12.

Action method

None.

2.3.5 Changing the Behavior of pgx_rcvall

Incompatibility

In FUJITSU Enterprise Postgres 13, change the pgx_rcvall command to fail if the -e option of the pgx_rcvall command specifies a future time or if the -n option specifies a list appointment that does not exist.

FUJITSU Enterprise Postgres 12 or earlier

All archived WALs are applied for recovery if the -e option of the pgx_rcvall command specifies a time in the future, or if the -n option specifies a list appointment that does not exist.

FUJITSU Enterprise Postgres 13 or later

The pgx_rcvall command fails if the -e option of the pgx_rcvall command specifies a time in the future, or if the -n option specifies a list appointment that does not exist.

Action method

Specify recovery objectives correctly, if necessary.

2.3.6 Mirroring Controller no Longer Retries to Monitor Database Processes when they are Detected as Down

Incompatibility

For the FUJITSU Enterprise Postgres Mirroring Controller, change the heartbeat monitoring of the database process so that it does not retry monitoring when it detects down.

Action method

None.

2.3.7 Changing the Name and Parameter Name of the Mirroring Controller Post-Promote Command

Incompatibility

In the FUJITSU Enterprise Postgres 12 Mirroring Controller, change the name of the post-promote command, which is the state transition command, and the parameter name in the server configuration file that specifies the post-promote command.

FUJITSU Enterprise Postgres 11 or earlier

- Command Name

  post-promote command

- The parameter name in the server configuration file that specifies the post-promote command

  post_promote_command

FUJITSU Enterprise Postgres 12 or later

- Command Name

  post-switch command

- The parameter name in the server configuration file that specifies the post-promote command

  post_switch_command

Action method

The post_promote_command parameter in the server configuration file continues to be available in FUJITSU Enterprise Postgres 12 or later.If specified in the server configuration file, it acts as a post-switch command.You cannot specify the post_promote_command and post_switch_command parameters at the same time.

2.3.8 Changing Mirroring Controller User Command Input Values

Incompatibility

Arguments (Fixed value: primarycenter) have been added to the following user commands:

- Fencing command of the database server
- Arbitration command

- Post-switch command
- Pre-detach command
- Post-attach command

Action method

If you are checking the number of arguments in a user command, increase the number of arguments by one.

2.3.9 Changing the Maximum Number of Connections per Server

Incompatibility

The maximum number of connections per server is changed from 262,143 to 65,535.

Action method

None.

2.3.10 Changing the Encryption Settings Using the ALTER TABLESPACE Statement

Incompatibility

The ALTER TABLESPACE statement can be used to change the tablespace encryption settings if data is not stored in the tablespace.

Action method

None.

2.3.11 Changing the Default Configuration of the Cluster System Using Database Multiplexing

Incompatibility

In FUJITSU Enterprise Postgres 10, a split brain will not occur if heartbeat monitoring using an admin network times out, so the default is changed to a system configuration requiring an arbitration server acting as a third party.

FUJITSU Enterprise Postgres 9.6 or earlier

A cluster system comprises two database servers.

FUJITSU Enterprise Postgres 10 or later

By default, a cluster system comprises two database servers and an arbitration server.

Action method

If selecting the same system configuration as that in FUJITSU Enterprise Postgres 9.6 or earlier, set the parameters below in the server configuration file to perform automatic degradation unconditionally when a heartbeat abnormality is detected during OS/server heartbeat monitoring.

- Parameter: heartbeat_error_action
- Value: fallback

2.3.12 Changing the Default Operation when mc_ctl Command Options are Omitted

Incompatibility

In FUJITSU Enterprise Postgres 10, the default operation when the mc_ctl command options below are omitted has been changed:

- The -f option during start mode
- The -w option during start mode

FUJITSU Enterprise Postgres 9.6 or earlier

- If the -f option is omitted, automatic switching and disconnection immediately after the startup of Mirroring Controller will not be enabled.
- If the -w option is omitted, the system will not wait for operations to finish.

FUJITSU Enterprise Postgres 10 or later

- Even if the -f option is omitted, automatic switching and disconnection immediately after the startup of Mirroring Controller will be enabled.
- Even if the -w option is omitted, the system will wait for operations to finish.

Action method

If selecting the same operation as that in FUJITSU Enterprise Postgres 9.6 or earlier, take the action below when executing start mode of the mc_ctl command.

- If the -f option is omitted, specify the -F option.
- If the -w option is omitted, specify the -W option.

2.3.13 Changing the Connection Settings when Mirroring Controller Connects to an Instance

Incompatibility

In FUJITSU Enterprise Postgres 10, Mirroring Controller changes the connection setting below when connecting to an instance to detect failure of each database element.

- Application name

FUJITSU Enterprise Postgres 9.6 or earlier

- The application name is an empty string.

FUJITSU Enterprise Postgres 10 or later

- The application name is 'mc_agent'.

Action method

There is no method for changing the application name.

Therefore, if there is an application that identifies a database connection session of Mirroring Controller with the application name being an empty string, modify the process so that identification is performed using 'mc_agent'.

2.3.14 Changing the Status Display of the Mirroring Controller Server

Incompatibility

In FUJITSU Enterprise Postgres 10, the condition for displaying an abnormality for the process (WAL receive process) that receives transaction logs has been changed in regard to the server status display performed in status mode of the mc_ctl command of Mirroring Controller.

- Role of the applicable server (host_role): standby (standby)
- Display item: DBMS process status (db_proc_status)

- Display content: abnormal (abnormal process name)
  The applicability for incompatibility is determined based on whether 'wal_receiver' is included in the abnormality process name.

FUJITSU Enterprise Postgres 9.6 or earlier

Displays an abnormality when Mirroring Controller detects that the WAL send process has stopped.

FUJITSU Enterprise Postgres 10 or later

Displays an abnormality when Mirroring Controller detects that a streaming replication connection has not been established with the primary server using the WAL receive process.

Action method

If there is an application that determines the WAL receive process status from the display content retrieved using status mode of the mc_ctl command, modify as below to determine at the same level of accuracy as previously. However, this is not recommended due to incompatibilities accompanying the improvement in accuracy.

- If 'wal_sender' is included in the abnormal process name displayed in the 'abnormal' db_proc_status item of the 'primary' host_role, it is determined that an abnormality has occurred in the WAL receive process.
- If 'wal_sender' is not included in the abnormal process name displayed in the 'abnormal' db_proc_status item of the 'primary' host_role, it is determined that an abnormality has not occurred in the WAL receive process.

2.3.15 Changing the Operation when the synchronous_standby_names Parameter is Changed during Database Multiplexing Operation

Incompatibility

In FUJITSU Enterprise Postgres 9.6 or earlier, Mirroring Controller periodically accesses the database to retrieve the synchronous_standby_names parameter value using the SHOW command, and automatically recovers if changes are detected, in case the user accidentally changes the synchronous_standby_names parameter in postgresql.conf during database multiplexing mode. However, because this process uses the CPU of the database server for redundancy and SQL statements are executed with high frequency, these processes are not executed by default in FUJITSU Enterprise Postgres 10.

FUJITSU Enterprise Postgres 9.6 or earlier

Mirroring Controller checks if the synchronous_standby_names parameter in postgresql.conf has been mistakenly changed by the user.

FUJITSU Enterprise Postgres 10 or later

By default, Mirroring Controller does not check if the synchronous_standby_names parameter in postgresql.conf has been mistakenly changed by the user.

Action method

If selecting the same operation as that in FUJITSU Enterprise Postgres 9.6 or earlier, set the parameter below in the server definition file.

- Parameter: check_synchronous_standby_names_validation
- Value: 'on'

2.3.16 Changing Masking Policy Definition for Unsupported Data Types

Incompatibility

The data masking feature of FUJITSU Enterprise Postgres is updated so that an error occurs when a masking policy is defined for a column of an unsupported data type (array type or timestamp with timezone type).

FUJITSU Enterprise Postgres 9.5

The definition of a masking policy for a column of array type or timestamp with timezone type ends normally, but an error occurs when accessing a column of a table for which the policy is defined.

FUJITSU Enterprise Postgres 9.6 or later

An error occurs when defining a masking policy for a column of array type or timestamp with timezone type.

Action method

If a policy that was defined in FUJITSU Enterprise Postgres 9.5 includes masking of a column of an unsupported data type (array type or timestamp with timezone type), perform one of the following in FUJITSU Enterprise Postgres 9.6 or later:

- Ignore the error if the masking policy contains columns of only these data types, or delete the policy with the pgx_drop_confidential_policy system administration function.
- If the masking policy target contains columns of these data types and a supported data type, delete the problematic columns with the pgx_alter_confidential_policy system function.

2.4 JDBC Driver Incompatibility

Item	Pre-migration version							
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14
Changing the targetServerType Value	Y	Y	Y	Y	Y	N	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.4.1 Changing the targetServerType Value

Incompatibility

In FUJITSU Enterprise Postgres 13, the value of targetServerType specified in the connection string was changed. Therefore, the previously used values are no longer available.

Action method

If you specified a value for targetServerType, change the value as follows:

Table 2.2 Specified values for the target server

Server Selection Order	FUJITSU Enterprise Postgres 12 or earlier	FUJITSU Enterprise Postgres 13
Primary Server	master	primary
Standby Server	slave	secondary
Prefer Standby Server	preferSlave	preferSecondary
Any	any	any

2.5 ODBC Driver Incompatibility

Item	Pre-migration version							
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14
Cannot specify prefer-read for target_session_attrs	Y	Y	Y	Y	Y	Y	Y	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.5.1 Cannot specify prefer-read for target_session_attrs

Incompatibility

FUJITSU Enterprise Postgres 14 users will not see the "prefer-read" radio button in the "Target_Session_Attrs" item of the data source option selection screen.

Action method

Select prefer-standby.

2.6 .NET Data Provider Incompatibility

Item	Pre-migration version							
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14
Changing the Specification Method and Specified Values for the Application Connection Switch Feature	Y	Y	Y	Y	Y	Y	Y	Y
Changing the Behavior of the Application Connection Switch Feature	Y	Y	Y	Y	Y	Y	Y	Y
Behavior Change when Multiple Host Information is Specified in the Connection String	Y	Y	Y	Y	Y	Y	Y	Y
Changing the TargetServerType Value	Y	Y	Y	Y	Y	N	N	N
Changing the Server Explorer View in Visual Studio Integration with .NET Data Provider	Y	Y	N	N	N	N	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.6.1 Changing the Specification Method and Specified Values for the Application Connection Switch Feature

Incompatibility

In Fujitsu Enterprise Postgres 15, the name of the "target server" specified by the connection switching function will be changed from "TargetServerType" to "TargetSessionAttributes", and the specified value will also be changed.

Action method

If the target server is specified, change the target server and specified value as follows.

Table 2.3 How to specify the connection destination switching function of the application and the specified value

Fujitsu Enterprise Postgres version		**FUJITSU Enterprise Postgres 9.6 or earlier**	**FUJITSU Enterprise Postgres 10/11/12**	**FUJITSU Enterprise Postgres 13/14**	**Fujitsu Enterprise Postgres 15**
Keywords to specify in the connection string		target_server	TargetServerType	TargetServerType	TargetSessionAttributes
Server selecti	Primary Server	primary(*1)	master(*1)	primary(*1)	read-write(*1) primary(*2)

Fujitsu Enterprise Postgres version		FUJITSU Enterprise Postgres 9.6 or earlier	FUJITSU Enterprise Postgres 10/11/12	FUJITSU Enterprise Postgres 13/14	Fujitsu Enterprise Postgres 15
on order	Standby Server	-	slave(*2)	standby(*2)	standby read-only(*2)
	Prefer Primary Server	-	-	-	prefer-primary
	Prefer Standby Server	prefer_standby	preferSlave	preferStandby	prefer-standby
	Any	-	any	any	any

*1: A primary server whose default transaction mode is read-only are not selected.

*2: A primary server whose default transaction mode is read-only is also selected.

2.6.2 Changing the Behavior of the Application Connection Switch Feature

Incompatibility

In Fujitsu Enterprise Postgres 15, when only one Host keyword is specified in the connection string, if a target server other than Any is specified, an exception will occur.

FUJITSU Enterprise Postgres 14 earlier

For example, if the connection string specifies "Host=host1, TargetServerType=primary", it will connect to host1 if host1 is the appropriate connection target.

Fujitsu Enterprise Postgres 15

For example, if "Host=host1, TargetSessionAttributes=primary" is specified in the connection string, an exception will always occur because the target server other than Any is specified for the specification of only one host information.

Action method

None.

2.6.3 Behavior Change when Multiple Host Information is Specified in the Connection String

Incompatibility

In Fujitsu Enterprise Postgres 15, when multiple hosts are specified in the connection string, even if one of them has an empty string specified, if another connectable connection destination is specified, that connection destination will be connected. In the old version, if there was even one host with an empty string, an exception would occur at that point.

FUJITSU Enterprise Postgres 14 earlier

For example, if "Host = host1," is specified in the connection string, an exception will always occur because the second host information is an empty string.

Fujitsu Enterprise Postgres 15

For example, if the connection string specifies "Host = host1," and if host1 is reachable, it will connect to host1 without throwing an exception.

Action method

None.

2.6.4 Changing the TargetServerType Value

Incompatibility

In FUJITSU Enterprise Postgres 13, the value of TargetServerType specified in the connection string was changed. Therefore, the previously used values are no longer available.

Action method

If you specified a value for TargetServerType, change the value as follows:

Table 2.4 Specified values for the target server

Server Selection Order	FUJITSU Enterprise Postgres 9.6 or earlier	FUJITSU Enterprise Postgres 10/11/12	FUJITSU Enterprise Postgres 13
Primary Server	primary	master	primary
Standby Server	-	slave	standby
Prefer Standby Server	prefer_standby	preferSlave	preferStandby
Any	-	any	any

2.6.5 Changing the Server Explorer View in Visual Studio Integration with .NET Data Provider

Incompatibility

FUJITSU Enterprise Postgres 10 or later, the following changes occur when you view database resources in Server Explorer.

- "Schema Name.Table Name" display under the Tables folder.
- The Indexes folder is not displayed.
- The Triggers folder does not displayed.
- The Sequences folder is not displayed.
- The Procedures folder does not displayed.

Action method

None.

2.7 C Library (libpq) Migration Incompatibility

Item	Pre-migration version							
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14
Changing when "prefer-read" is Specified for the target_session_attrs Parameter	Y	Y	Y	Y	Y	Y	Y	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.7.1 Changing when "prefer-read" is Specified for the target_session_attrs Parameter

Incompatibility

In FUJITSU Enterprise Postgres 14 changes the attach server priority if any of the following servers are specified simultaneously with "prefer-read" as the target_session_attrs parameter:

- Primary server (default_transaction_read_only = ON)
- Standby server

FUJITSU Enterprise Postgres 13 SP1 or earlier

The primary server (default_transaction_read_only = ON) and standby servers have the same priority.

FUJITSU Enterprise Postgres 14 or later

Standby servers connect in preference to primary servers (default_transaction_read_only = ON).

Action method

None.

2.8 pgaudit Incompatibility

Item	Pre-migration version							
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14
Changing to Output Extra NEW and OLD Values in the Audit Log when the Trigger Function Executes	N	N	Y	Y	Y	N	N	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.8.1 Changing to Output Extra NEW and OLD Values in the Audit Log when the Trigger Function Executes

Incompatibility

In FUJITSU Enterprise Postgres 13, the trigger function additionally outputs NEW and OLD values to the audit log when the pgaudit.log_parameter is set to on.

FUJITSU Enterprise Postgres 12 or earlier

The following (18) does not output the values of NEW, OLD.

[Example]

```
AUDIT: SESSION,WRITE,2020-09-03 07:07:39 UTC,
        (1)    (2)   (3)
[local],9775,psql,k5user,postgres,3/536, 1, 2, INSERT, ,
(4)     (5)  (6)  (7)    (8)      (9)   (10)(11)(12)(13)
TABLE,public.trig_audit, ,
(14)  (15)    (16)
 "INSERT INTO trig_audit SELECT 'U', now(), user, OLD.*, NEW.*",
(17)
trig_audit AFTER ROW UPDATE 92027 trig_test trig_test public 0  f aaaa
(18)
```

FUJITSU Enterprise Postgres 13 or later

NEW, OLD values are output.

[Example]

```
AUDIT: SESSION,WRITE,2020-09-03 07:07:39 UTC,
[local],9775,psql,k5user,postgres,3/536,1,2,INSERT,,TABLE,public.
trig_audit,,
"INSERT INTO trig_audit SELECT 'U', now(), user, OLD.*, NEW.*",
(bbb) (aaa) trig_audit AFTER ROW UPDATE 92027 trig_test trig_test
public 0  f aaaa
```

Action method

If you are using an application that works by monitoring the string that the trigger function output to the audit log, modify the application to work with the NEW and OLD values.

2.9 WebAdmin Incompatibility

Item	Pre-migration version							
	9.5	9.6	10	11	12	13	12 SP1/13 SP1	14
Cannot specify prefer-read for target_session_attrs	Y	Y	Y	Y	Y	Y	Y	N

Y: Incompatibility exists

N: Incompatibility does not exist

2.9.1 Cannot specify prefer-read for target_session_attrs

Incompatibility

FUJITSU Enterprise Postgres 14 will no longer allow prefer-read to target_session_attrs as a connection method to an upstream server that is specified when creating an instance of a standby server.

Action method

Specify prefer-standby.

Chapter 3 Program Updates

For program fix information, see "Program Updates".

Index

[C]

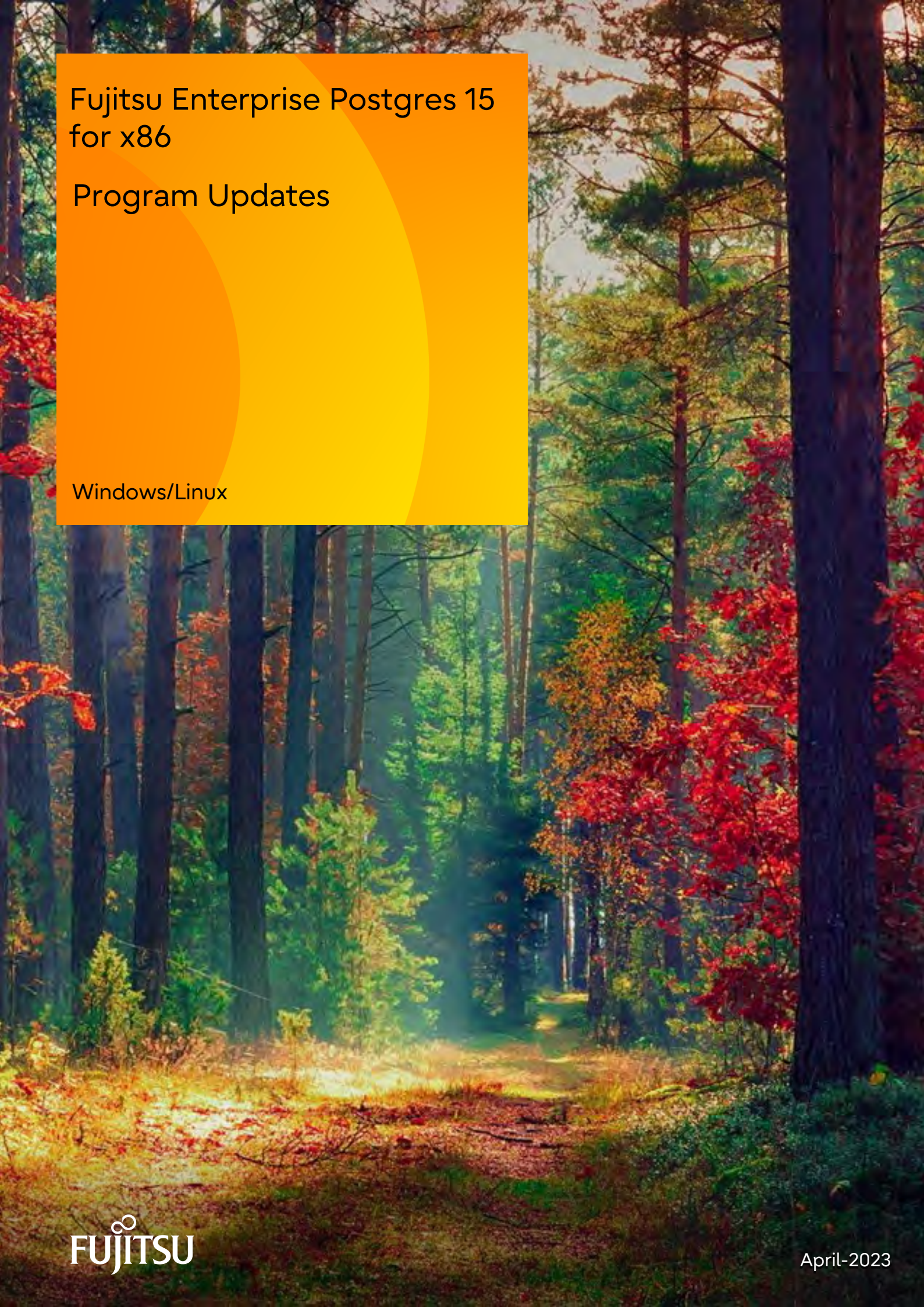
Compatibility Information.....4

[F]

Features Added in 15.....1

[P]

Program Updates.....27



Fujitsu Enterprise Postgres 15 for x86

Program Updates

Windows/Linux

FUJITSU

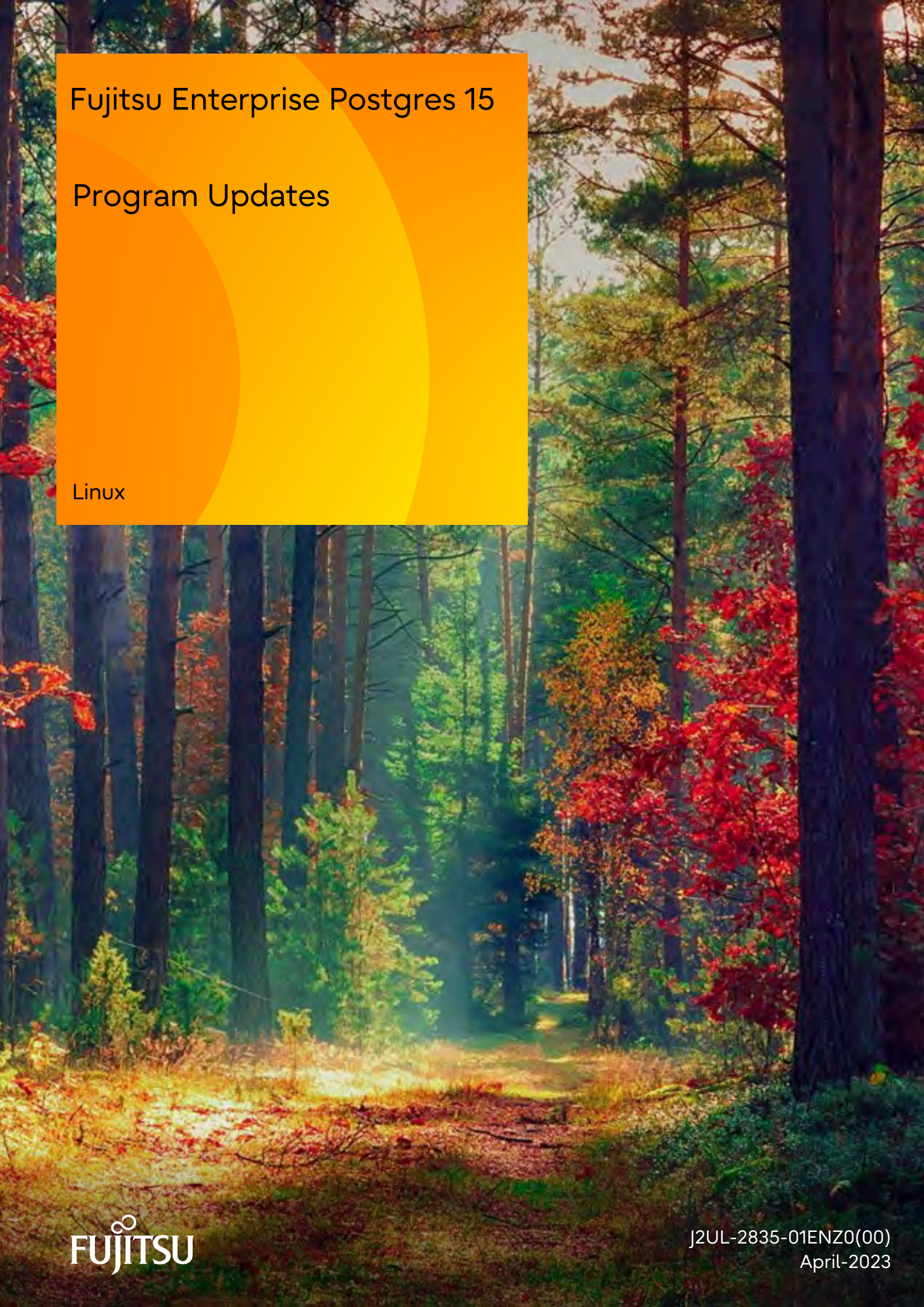
April-2023

Linux >
Windows >

Fujitsu Enterprise Postgres 15

Program Updates

Linux



FUJITSU

J2UL-2835-01ENZ0(00)
April-2023

Preface

This document explains the updates that have been fixed at this version.

The contents of this document are subject to change without notice.

Notations

The status for each edition is shown in the following table.

P number	Update summary	AE	SE
Number that uniquely identifies the update	Summary of update details	Symbol indicating the fixed status	

AE

Indicates Advanced Edition.

SE

Indicates Standard Edition.

Symbol

Indicates the incorporated status for each edition.

Y: Fixed

-: Not relevant to this product

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

```
Edition 1.0: April  2023
```

Copyright

Copyright 2020-2023 Fujitsu Limited

Chapter 1 Program Updates

This version incorporates the updates implemented in PostgreSQL 15.

See

Refer to the PostgreSQL Global Development Group website for information on the updates implemented in the following releases:

[PostgreSQL 15]

```
https://www.postgresql.org/docs/15/release-15.html
```

In addition, issues that occurred in previous versions are also fixed.

Refer to the following for details of the program fixes included in this version and level.

- Fujitsu Enterprise Postgres 15 Program Updates

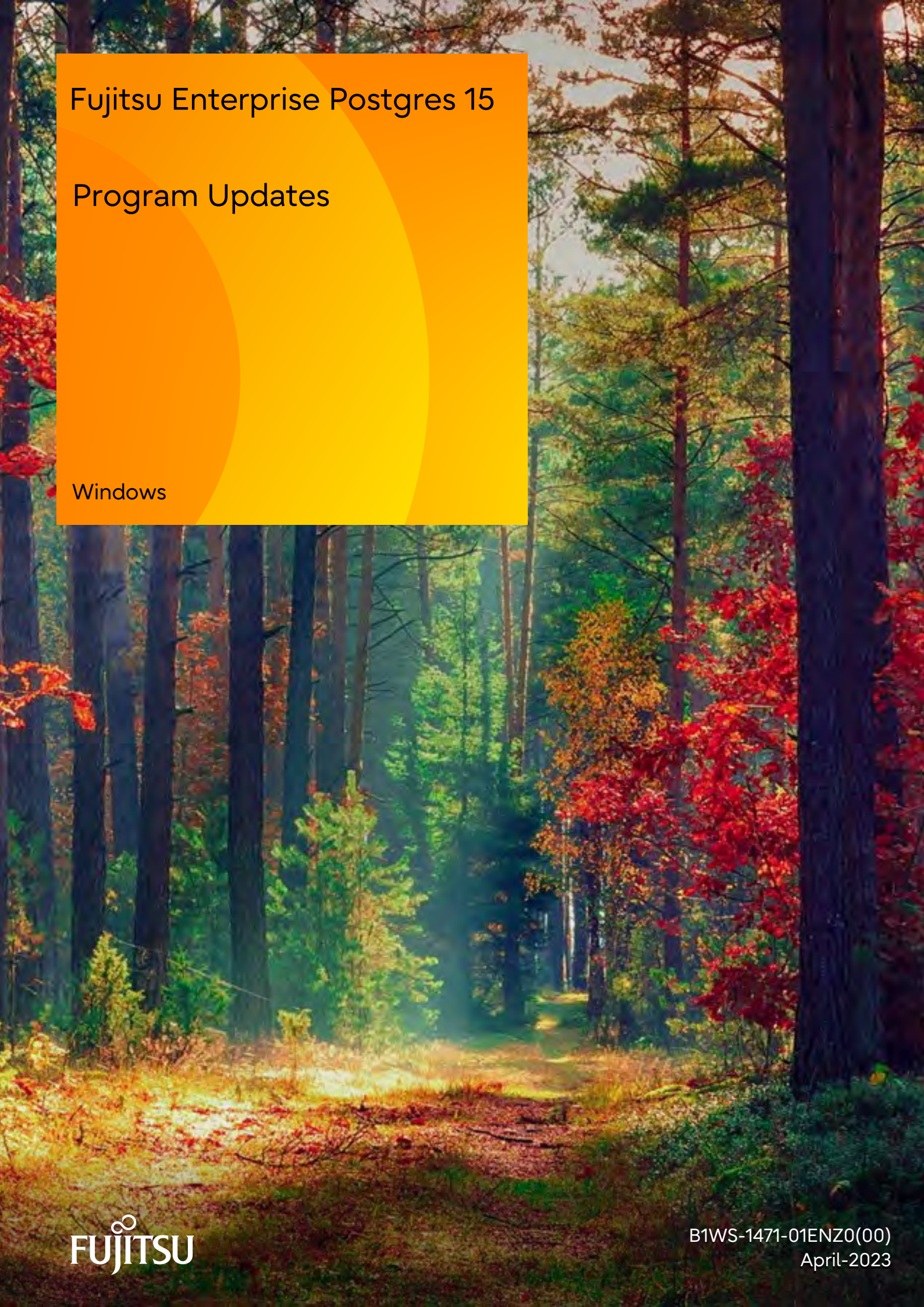
Fujitsu Enterprise Postgres 15 Program Updates

P number	Update summary	AE	SE
PH16261	When deleting a synchronous standby instance using WebAdmin, the application name remains in the parameter synchronous_standby_names of the master instance.	Y	Y
PH18843	An unnecessary WARNING message may be output when the transaction log duplication feature is enabled.	Y	Y
PH19098	When using pgaudit, an SQL error may occur.	Y	-
PH19668	When ODBC driver's advanced configuration in the ODBC Data Source Administrator is performed, the MyLog settings are automatically enabled and cannot be disabled.	Y	Y
PH19947	After a database switch occurs, applications connected using the Connection Manager may not be able to reconnect to the database.	Y	-
PH20434	Metacache may not be placed in the GMC area on the standby server.	Y	-
PH20625	Using the features compatible with Oracle database may result in an error.	Y	Y
PH21161	When Mirroring Controller detects an OS error, automatic switching or automatic disconnection may fail after executing the arbitration server fencing command.	Y	Y
PH21305	When executing the mc_ctl status command, the status may become unknown depending on the network and CPU conditions.	Y	Y
PH21560	A new parameter audit_log_disconnections is added to output disconnections in the CONNECT class (events related to connections) to the audit log.	Y	-
PH21759	PRIMECLUSTER linkage cannot be started in an environment using a login shell other than bash.	Y	Y
PH22445	When executing SQL statement or VACUUM on a table with a GIN index, the incorrect scan result may return or the server process may be abort.	Y	Y
PH22581	When creating the table which inludes the column of national character data type by using pgAdmin4, the data length of the column may be not able to be specified.	Y	Y
PH22587	When inserting or updating the column data of national character data type by using pgAdmin4, the data which is unexpected length may be inserted or updated.	Y	Y
PH22603	When using a key management system as a key store for transparent data encryption, if the master encryption key is changed, it may fail to apply WAL.	Y	Y
PH22649	When updating the data type with national character data type by using pgAdmin4, the column may be updated the unexpected length.	Y	Y
PH22741	When started in single-user mode, the postgres process may core dump and terminate abnormally.	Y	Y

Fujitsu Enterprise Postgres 15

Program Updates

Windows



FUJITSU

B1WS-1471-01ENZ0(00)
April-2023

Preface

This document explains the updates that have been fixed at this version.

The contents of this document are subject to change without notice.

Notations

The status for each edition is shown in the following table.

P number	Update summary	AE	SE
Number that uniquely identifies the update	Summary of update details	Symbol indicating the fixed status	

AE

Indicates Advanced Edition.

SE

Indicates Standard Edition.

Symbol

Indicates the incorporated status for each edition.

Y: Fixed

-: Not relevant to this product

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

```
Edition 1.0: April 2023
```

Copyright

Copyright 2020-2023 Fujitsu Limited

Chapter 1 Program Updates

This version incorporates the updates implemented in PostgreSQL 15.

See

Refer to the PostgreSQL Global Development Group website for information on the updates implemented in the following releases:

[PostgreSQL 15]

```
https://www.postgresql.org/docs/15/release-15.html
```

In addition, issues that occurred in previous versions are also fixed.

Refer to the following for details of the program fixes included in this version and level.

- Fujitsu Enterprise Postgres 15 Program Updates

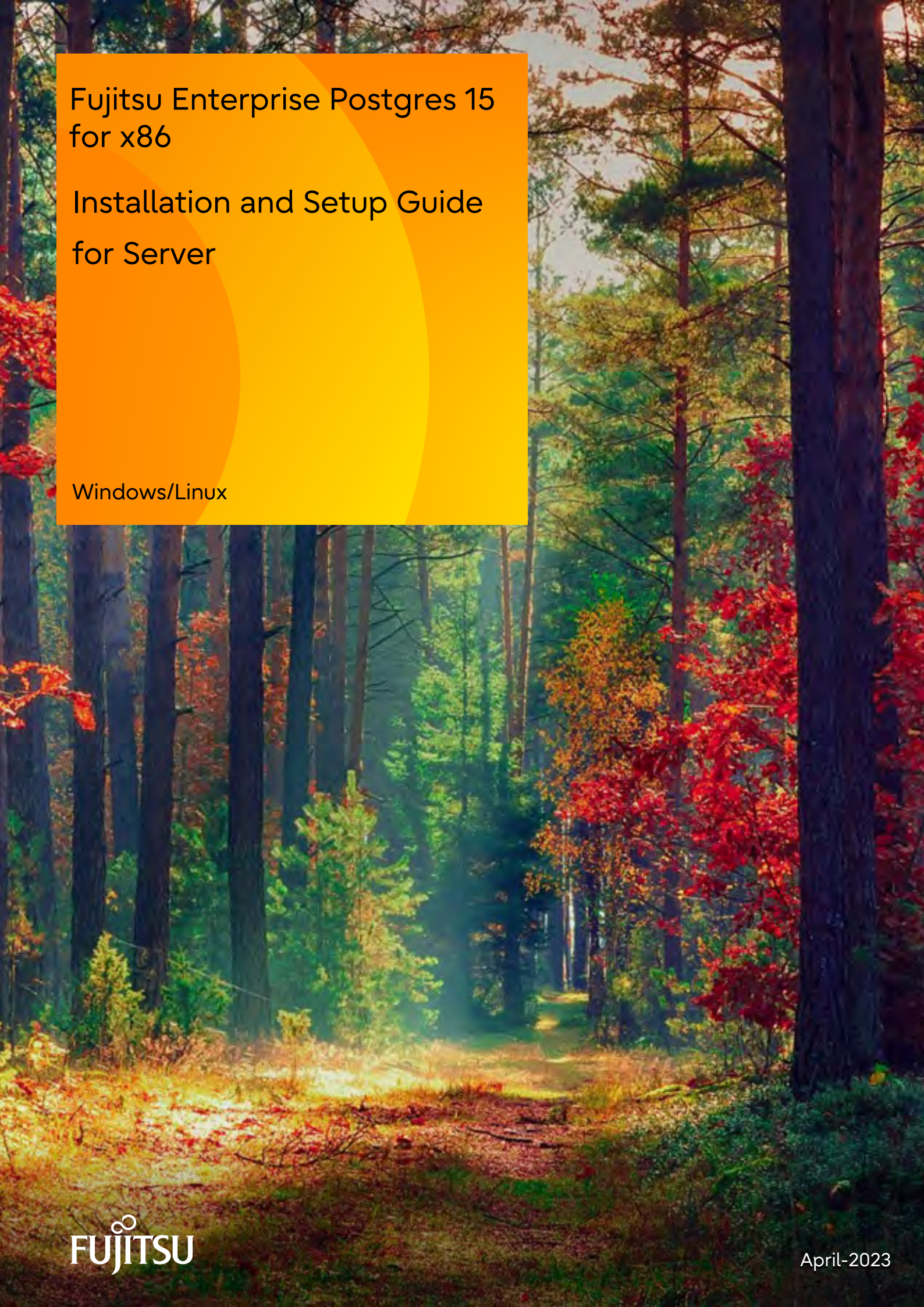
Fujitsu Enterprise Postgres 15 Program Updates

P number	Update summary	AE	SE
PH16261	When deleting a synchronous standby instance using WebAdmin, the application name remains in the parameter synchronous_standby_names of the master instance.	Y	Y
PH18843	An unnecessary WARNING message may be output when the transaction log duplication feature is enabled.	Y	Y
PH19098	When using pgaudit, an SQL error may occur.	Y	-
PH19668	When ODBC driver's advanced configuration in the ODBC Data Source Administrator is performed, the MyLog settings are automatically enabled and cannot be disabled.	Y	Y
PH19947	After a database switch occurs, applications connected using the Connection Manager may not be able to reconnect to the database.	Y	-
PH20434	Metacache may not be placed in the GMC area on the standby server.	Y	-
PH20625	Using the features compatible with Oracle database may result in an error.	Y	Y
PH21161	When Mirroring Controller detects an OS error, automatic switching or automatic disconnection may fail after executing the arbitration server fencing command.	Y	-
PH21305	When executing the mc_ctl status command, the status may become unknown depending on the network and CPU conditions.	Y	-
PH21560	A new parameter audit_log_disconnections is added to output disconnections in the CONNECT class (events related to connections) to the audit log.	Y	-
PH22445	When executing SQL statement or VACUUM on a table with a GIN index, the incorrect scan result may return or the server process may be abort.	Y	Y
PH22581	When creating the table which inludes the column of national character data type by using pgAdmin4, the data length of the column may be not able to be specified.	Y	Y
PH22587	When inserting or updating the column data of national character data type by using pgAdmin4, the data which is unexpected length may be inserted or updated.	Y	Y
PH22649	When updating the data type with national character data type by using pgAdmin4, the column may be updated the unexpected length.	Y	Y
PH22741	When started in single-user mode, the postgres process may core dump and terminate abnormally.	Y	Y

Fujitsu Enterprise Postgres 15 for x86

Installation and Setup Guide for Server

Windows/Linux



FUJITSU

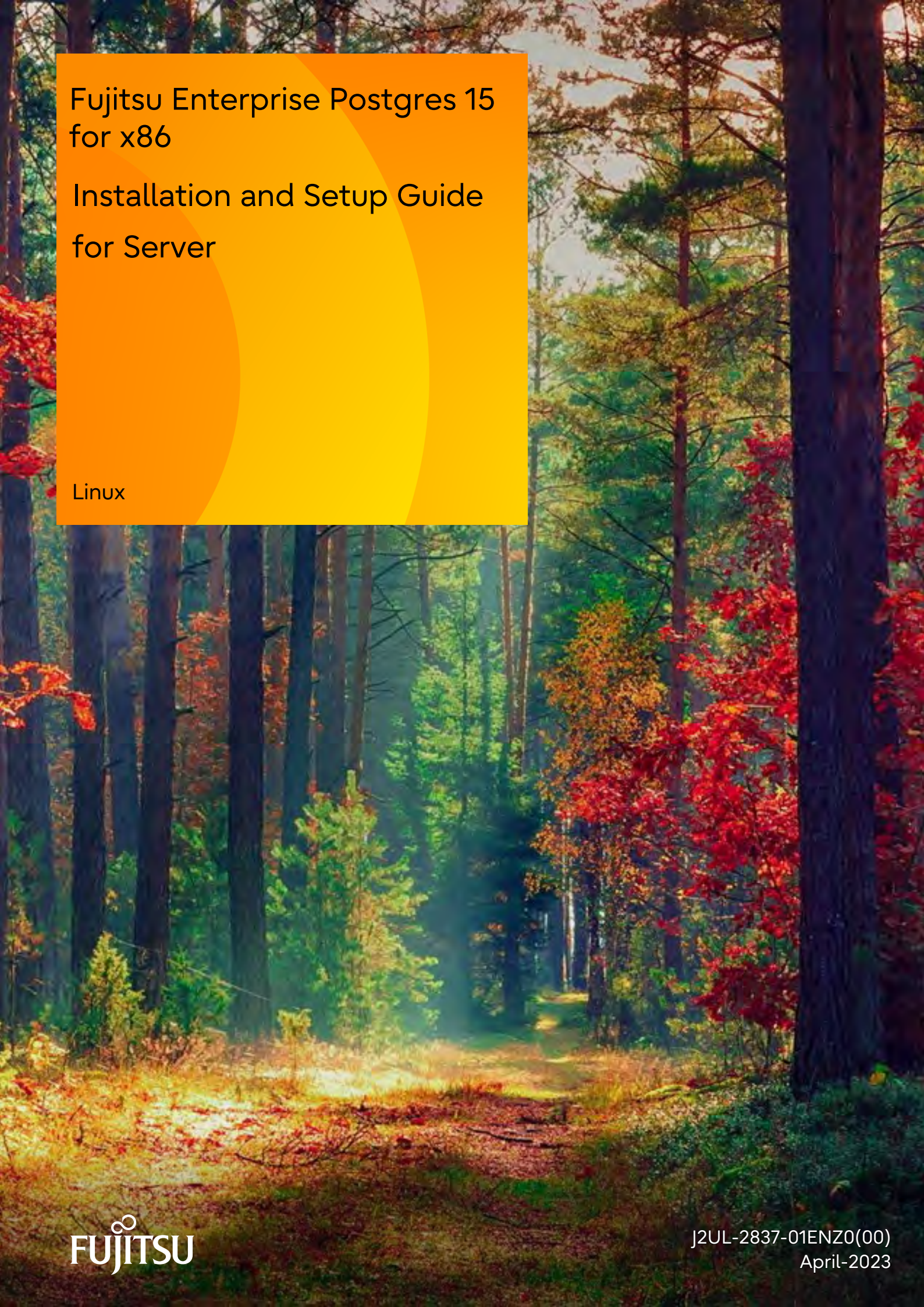
April-2023

Linux >
Windows >

Fujitsu Enterprise Postgres 15 for x86

Installation and Setup Guide for Server

Linux



FUJITSU

J2UL-2837-01ENZ0(00)
April-2023

Preface

Purpose of this document

The Fujitsu Enterprise Postgres database system extends the PostgreSQL features and runs on the Linux platform.

This document describes how to install and set up "Fujitsu Enterprise Postgres".

Intended readers

This document is intended for those who install and operate Fujitsu Enterprise Postgres.

Readers of this document are assumed to have general knowledge of:

- PostgreSQL
- SQL
- Linux

Structure of this document

This document is structured as follows:

Chapter 1 Overview of Installation

Describes the installation types and procedures

Chapter 2 Operating Environment

Describes the operating environment required to use Fujitsu Enterprise Postgres

Chapter 3 Installation

Describes how to perform a new installation of Fujitsu Enterprise Postgres

Chapter 4 Setup

Describes the setup to be performed after installation

Chapter 5 Uninstallation

Describes how to uninstall Fujitsu Enterprise Postgres

Appendix A Recommended WebAdmin Environments

Describes the recommended WebAdmin environment.

Appendix B Setting Up and Removing WebAdmin

Describes how to set up and remove WebAdmin

Appendix C WebAdmin Disallow User Inputs Containing Hazardous Characters

Describes characters that are not allowed in WebAdmin.

Appendix D Configuring Parameters

Describes Fujitsu Enterprise Postgres parameters.

Appendix E Estimating Database Disk Space Requirements

Describes how to estimate database disk space requirements

Appendix F Estimating Memory Requirements

Describes the formulas for estimating memory requirements

Appendix G Quantitative Limits

Describes the quantity range

Appendix H Configuring Kernel Parameters

Describes the settings for kernel parameters

Appendix I Determining the Preferred WebAdmin Configuration

Describes the two different configurations in which WebAdmin can be used and how to select the most suitable configuration

Appendix J System Configuration when using Pgpool-II

Describes the system configuration when using Pgpool-II.

Appendix K Supported contrib Modules and Extensions Provided by External Projects

Lists the PostgreSQL contrib modules and the extensions provided by external projects supported by Fujitsu Enterprise Postgres.

Appendix L Procedure when Modifying the JRE Installation

Describes the procedures to follow when modifying the JRE installation.

Appendix M Access to Key Management System Using Plug-in

Describes how to access key management systems using plug-ins.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version


```
Edition 1.0:  April 2023
```


Copyright

Copyright 2015-2023 Fujitsu Limited

Contents

Chapter 1 Overview of Installation....1
1.1 Features that can be Installed....1
1.2 Installation Types....1
1.2.1 New Installation....1
1.2.2 Reinstallation....1
1.2.3 Multi-Version Installation....1
1.3 Installation Procedure....1
1.3.1 Installation in Interactive Mode....2
1.3.2 Installation in Silent Mode....2
1.4 Uninstallation....2

Chapter 2 Operating Environment....3
2.1 Required Operating System....3
2.2 Related Software....9
2.3 Excluded Software....10
2.4 Required Patches....11
2.5 Hardware Environment....11
2.6 Disk Space Required for Installation....11
2.7 Supported System Environment....12
2.7.1 TCP/IP Protocol....12
2.7.2 File System....12
2.8 PostgreSQL Version Used for Fujitsu Enterprise Postgres....12
2.9 Notes on Using Streaming Replication....12
2.10 Key Management System Requirements....12
2.10.1 To Connect to a key Management System Using the KMIP Protocol....12
2.10.2 To Connect to a Key Management System Using a Plug-in....13

Chapter 3 Installation....14
3.1 Pre-installation Tasks....14
3.2 Installation in Interactive Mode....16
3.3 Installation in Silent Mode....18

Chapter 4 Setup....20
4.1 Operating Method Types and Selection....20
4.2 Preparations for Setup....21
4.2.1 Creating an Instance Administrator....21
4.2.2 Preparing Directories for Resource Deployment....22
4.2.3 Estimating Resources....25
4.2.4 Configuring Corefile Names....25
4.3 Creating Instances....25
4.3.1 Using WebAdmin....26
4.3.1.1 Logging in to WebAdmin....26
4.3.1.2 Creating an Instance....27
4.3.1.3 Changing Instance Settings....30
4.3.1.4 Importing Instances....32
4.3.2 Using the initdb Command....33
4.3.2.1 Editing Kernel Parameters....33
4.3.2.2 Creating an Instance....33
4.4 Configuring Remote Connections....36
4.4.1 When an Instance was Created with WebAdmin....36
4.4.2 When an Instance was Created with the initdb Command....36
4.5 Other Settings....37
4.5.1 Error Log Settings....37
4.5.2 Configuring Automatic Start and Stop of an Instance....38
4.5.3 Settings when Using the Features Compatible with Oracle Databases....39
4.5.4 LDAP Authentication File Settings....39

4.5.5 Setting the server keytab file for GSSAPI authentication ... 39
4.6 Integration with Message-Monitoring Software ... 39
4.7 Setting Up and Removing OSS ... 40
4.7.1 oracle_fdw ... 40
4.7.1.1 Setting Up oracle_fdw ... 40
4.7.1.2 Removing oracle_fdw ... 41
4.7.2 pg_bigm ... 41
4.7.2.1 Setting Up pg_bigm ... 41
4.7.2.2 Removing pg_bigm ... 41
4.7.3 pg_hint_plan ... 42
4.7.3.1 Setting Up pg_hint_plan ... 42
4.7.3.2 Removing pg_hint_plan ... 42
4.7.4 pg_dbms_stats ... 43
4.7.4.1 Setting Up pg_dbms_stats ... 43
4.7.4.2 Removing pg_dbms_stats ... 43
4.7.5 pg_repack ... 44
4.7.5.1 Setting Up pg_repack ... 44
4.7.5.2 Removing pg_repack ... 44
4.7.6 pg_rman ... 45
4.7.6.1 Setting Up pg_rman ... 45
4.7.6.2 Removing pg_rman ... 45
4.7.7 pg_statsinfo ... 45
4.7.7.1 Setting Up pg_statsinfo ... 45
4.7.7.2 Removing pg_statsinfo ... 46
4.7.8 pgBadger ... 46
4.7.8.1 Setting Up pgBadger ... 46
4.7.8.2 Removing pgBadger ... 46
4.7.9 Pgpool-II ... 47
4.7.9.1 Setting Up Pgpool-II ... 47
4.7.9.2 Removing Pgpool-II ... 47
4.7.10 Build with PGXS ... 47
4.7.10.1 Using the Default Version of llvm ... 48
4.7.10.2 Using a Non-Default Version of llvm ... 48
4.7.10.3 Without llvm ... 48
4.7.10.4 Setting DT_RUNPATH ... 48
4.7.11 Build without PGXS ... 48
4.8 Deleting Instances ... 49
4.8.1 Using WebAdmin ... 49
4.8.2 Using Server Commands ... 49

Chapter 5 Uninstallation ... 51
5.1 Uninstallation in Interactive Mode ... 51
5.2 Uninstallation in Silent Mode ... 53

Appendix A Recommended WebAdmin Environments ... 55
A.1 Recommended Browser Settings ... 55
A.2 How to Set Up the Pop-up Blocker ... 55

Appendix B Setting Up and Removing WebAdmin ... 56
B.1 Setting Up WebAdmin ... 56
B.1.1 Setting Up WebAdmin ... 56
B.1.2 Starting the Web Server Feature of WebAdmin ... 57
B.1.3 Stopping the Web Server Feature of WebAdmin ... 57
B.2 Removing WebAdmin ... 58
B.3 Using an External Repository for WebAdmin ... 58
B.4 Using the WebAdmin Auto-Refresh Feature ... 60

Appendix C WebAdmin Disallow User Inputs Containing Hazardous Characters ... 61

Appendix D Configuring Parameters....62

Appendix E Estimating Database Disk Space Requirements....66
E.1 Estimating Table Size Requirements....66
E.2 Estimating Index Size Requirements....67
E.3 Sizes of Data Types....68
E.3.1 Sizes of Fixed-Length Data Types....68
E.3.2 Sizes of Variable-Length Data Types....69
E.3.3 Sizes of Array Data Types....69
E.3.4 Number of Bytes per Character....70
E.4 Estimating Transaction Log Space Requirements....70
E.5 Estimating Archive Log Space Requirements....70
E.6 Estimating Backup Disk Space Requirements....70
E.7 Estimating VCI Disk Space Requirements....70

Appendix F Estimating Memory Requirements....72
F.1 Fujitsu Enterprise Postgres Memory Requirements....72
F.2 Database Multiplexing Memory Requirements....73
F.3 VCI Memory Requirements....73
F.4 High-Speed Data Load Memory Requirements....75
F.5 Global Meta Cache Memory Requirements....75

Appendix G Quantitative Limits....76

Appendix H Configuring Kernel Parameters....81

Appendix I Determining the Preferred WebAdmin Configuration....82
I.1 WebAdmin Configurations....82
I.1.1 Single-Server Configuration....82
I.1.2 Multiserver Configuration....82
I.2 Installing WebAdmin in a Single-Server Configuration....83
I.3 Installing WebAdmin in a Multiserver Configuration....84

Appendix J System Configuration when using Pgpool-II....85
J.1 Pgpool-II Configuration....85
J.1.1 Single-Machine Configuration....85
J.1.2 Two-Machine Configuration....86
J.1.3 Three-Machine Configuration....86
J.2 Installing Pgpool-II....86
J.3 Pgpool-II Setup....87
J.3.1 Setting Environment Variables....87
J.3.2 Configuration file....87
J.3.2.1 Configuring pgpool.conf....87
J.3.2.2 Using Configuration Files....87

Appendix K Supported contrib Modules and Extensions Provided by External Projects....88

Appendix L Procedure when Modifying the JRE Installation....89
L.1 When Using WebAdmin....89
L.2 When Performing Database Multiplexing....89

Appendix M Access to Key Management System Using Plug-in....91
M.1 What to do with Plug-ins....91
M.2 Where the Plug-in is Stored....91
M.3 Invoking the Plug-in....91
M.4 Passing Confidential Information to Plug-ins....91
M.5 Calling Convention....91
M.5.1 Key Verification....91
M.5.2 Encryption....92

M.5.3 Decryption.....93

Index.....95

Chapter 1 Overview of Installation

This chapter provides an overview of Fujitsu Enterprise Postgres installation.

1.1 Features that can be Installed

Each Fujitsu Enterprise Postgres feature is installed on the machine that was used to build the database environment.

The following table shows the relationship between the product to be installed and the features that can be installed.

Feature that can be installed	Product name	
	AE	SE
Basic feature (server feature, client feature)	Y	Y

Y: Can be installed

1.2 Installation Types

The following installation types are available for Fujitsu Enterprise Postgres:

- New installation
- Reinstallation
- Multi-version installation

1.2.1 New Installation

In initial installation, Fujitsu Enterprise Postgres is installed for the first time.

1.2.2 Reinstallation

Perform reinstallation to repair installed program files that have become unusable for any reason.

1.2.3 Multi-Version Installation

Fujitsu Enterprise Postgres products can be installed on the same server if the product version (indicated by "*x*" in "*x* SP*z*") is different from that of any version of the product that is already installed.

1.3 Installation Procedure

The following installation procedures are available for Fujitsu Enterprise Postgres:

- Installation in interactive mode
- Installation in silent mode

Select the installation procedure that corresponds to your environment.

Note

If you have antivirus software installed, the server may crash, fail to start, or stop responding, during installation or when starting up after installation. Set scan exception settings for the installation directory and resource allocation directory so that the files in these directories are not scanned for viruses.

1.3.1 Installation in Interactive Mode

Interactive mode enables installation to be performed while the required information is entered interactively.

In the interactive mode installation, the installation state of Fujitsu Enterprise Postgres is determined automatically. Install Fujitsu Enterprise Postgres using one of the following installation types in accordance with the installation state:

- New installation
- Reinstallation
- Multi-version installation

1.3.2 Installation in Silent Mode

Silent mode enables installation to be performed without the need to enter any information interactively.

New installations and multi-version installations can be performed in silent mode.

1.4 Uninstallation

Uninstallation removes the system files of the installed Fujitsu Enterprise Postgres.

Chapter 2 Operating Environment

This chapter describes the operating environment required to use Fujitsu Enterprise Postgres.

See

Refer to "Operating Environment" in the Installation and Setup Guide for Client when installing the Fujitsu Enterprise Postgres client feature at the same time.

2.1 Required Operating System

One of the operating systems shown below is required in order to use Fujitsu Enterprise Postgres.

- RHEL7.4 or later minor version
- RHEL8.2 or later minor version
- RHEL9.0 or later minor version
- SLES 12 SP5
- SLES 15 SP3 or later minor version

Note

SLES cannot be used if performing failover operation integrated with PRIMECLUSTER.

Information

- The following packages are required for operations on RHEL7.

Package name	Remarks
alsa-lib	-
audit-libs	-
dstat	Required when using parallel scan.
gdb	Required to run FJQSS.
glibc	-
glibc.i686	-
iputils	Required for Mirroring Controller.
libgcc	-
libicu	Provides collation support. Install 50.x.
libmemcached	Required when using Pgpool-II.
libselinux	Required for sepgsql.
libstdc++	-
libtool-ltdl	Required when using ODBC drivers.
libzstd	-
llvm	Version 10.0.x of llvm is required to run SQL with runtime compilation (just-in-time compilation). Install the package that

Package name	Remarks
	contains libLLVM-10.so.For example, the "llvm-toolset-10.0-llvm-libs" published in Red Hat Developer Tools and Red Hat Software Collections includes libLLVM10.so. Fujitsu Enterprise Postgres uses runtime compilation by default.If you do not want to use runtime compilation, turn off the jit parameter in postgresql.conf.You do not need to install llvm if you turn off the jit parameter. Failure to install llvm without turning off the jit parameter may result in an error when executing SQL.For more information about runtime compilation, see "Just-in-Time Compilation (JIT)" in "PostgreSQL Documentation".
lz4	-
ncurses-libs	-
net-tools	-
nss-softokn-freebl	-
pam	Required when using PAM authentication.
perl-libs	Required when using PL/Perl. Install 5.16.
protobuf-c	Required if using the Transparent Data Encryption feature when using a key management system as a keystore. Install 1.0.2.
python3	Required when using PL/Python based on Python 3. Install 3.6.x .
redhat-lsb	-
rsync	Required when using Pgpool-II.
sysstat	Required when using FJQSS. Set up the sar command after installation.
tcl	Required when using PL/Tcl. Install 8.5.
unzip	-
xz-libs	-
zlib	-

- The following packages are required for operations on RHEL8.

Package name	Remarks
alsa-lib	-
audit-libs	-
cyrus-sasl-lib	-
gdb	Required to run FJQSS.
pcp-system-tools	Required when using parallel scan.
glibc	-
glibc.i686	-
iputils	Required for Mirroring Controller.
libnsl2	-

Package name	Remarks
libicu	Provides collation support. Install 60.x.
libgcc	-
libmemcached	Required when using Pgpool-II.
libselinux	Required for sepgsql.
libstdc++	-
libtool-ltdl	Required when using ODBC drivers.
libzstd	-
llvm	Vversions 13.0.x, 12.0.x, 11.0.x, 10.0.x, or 9.0.x of llvm is required to run SQL with runtime compilation (just-in-time compilation). Install the package that contains libLLVM-13.so, libLLVM-12.so, libLLVM-11.so. libLLVM-10.so, or libLLVM-9.so. For example, version 13.0.x of "llvm-libs" published with Application Streams includes libLLVM-13.so. By default, version 13.0.x is used. If you use a version other than 13.0.x, specify the version you want to use in the jit_provider parameter in postgresql.conf. For example, use llvmjit-vsn12 when using version 12.0.x. Fujitsu Enterprise Postgres uses runtime compilation by default.If you do not want to use runtime compilation, turn off the jit parameter in postgresql.conf.You do not need to install llvm if you turn off the jit parameter. Failure to install llvm without turning off the jit parameter may result in an error when executing SQL.For more information about runtime compilation, see "Just-in-Time Compilation (JIT)" in "PostgreSQL Documentation".
lz4-libs	-
ncurses-libs	-
net-tools	-
nss-softokn-freebl	-
pam	Required when using PAM authentication.
perl-libs	Required when using PL/Perl. Install 5.26.
protobuf-c	Required if using the Transparent Data Encryption feature when using a key management system as a keystore. Install 1.3.0.
python3	Required when using PL/Python based on Python 3. Install 3.6.x .
redhat-lsb	-
rsync	Required when using Pgpool-II.
sysstat	Required when using FJQSS. Set up the sar command after installation.
tcl	Required when using PL/Tcl. Install 8.6.

Package name	Remarks
unzip	-
xz-libs	-
zlib	-

- The following packages are required for operations on RHEL9.

Package name	Remarks
alsa-lib	-
audit-libs	-
cyrus-sasl-lib	-
gdb	Required to run FJQSS.
pcp-system-tools	Required when using parallel scan.
glibc	-
glibc.i686	-
iputils	Required for Mirroring Controller.
libnsl2	-
libicu	Provides collation support. Install 67.x.
libgcc	-
libmemcached-awesome	Required when using Pgpool-II.
libselinux	Required for sepgsql.
libstdc++	-
libtool-ltdl	Required when using ODBC drivers.
libzstd	-
llvm	llvm version 13.0 .x is required to run SQL using just-in-time compilation. Install the package that contains libLLVM -13 .so. For example, version 13.0 .x of "llvm-libs " published with Application Streams includes libLLVM -13 .so. Fujitsu Enterprise Postgres is configured to use runtime compilation by default. If you do not want to use runtime compilation, turn off the jit parameter in postgresql.conf. If you turn off the jit parameter, you do not need to install llvm. Failure to install llvm without turning off the jit parameter can result in errors during SQL execution. For more information about runtime compilation, see "Just-in-Time Compilation (JIT) " in the PostgreSQL Documentation.
lz4-libs	-
ncurses-libs	-
net-tools	-
nss-softokn-freebl	-
pam	Required when using PAM authentication.

Package name	Remarks
perl-libs	Required when using PL/Perl. Install 5.32.
protobuf-c	Required if using the Transparent Data Encryption feature when using a key management system as a keystore. Install 1.3.3.
python3	Required when using PL/Python based on Python 3. Install 3.9.x .
rsync	Required when using Pgpool-II.
sysstat	Required when using FJQSS. Set up the sar command after installation.
tcl	Required when using PL/Tcl. Install 8.6.
unzip	-
xz-libs	-
zlib	-

- The following packages are required for operations on SLES 12.

Package name	Remarks
dstat	Required when using parallel scan.
gdb	Required to run FJQSS.
glibc	-
glibc-32bit	-
iputils	Required for Mirroring Controller.
JRE 8	Required when using the installer, database multiplexing, and WebAdmin. The following JREs are available: - Oracle JRE Use update 31 or later of the 64-bit version. - OpenJDK Past the Java TCK (Technology Compatibility Kit)
libasound2	-
libaudit1	-
libfreebl3	-
libicu	Provides collation support. Install 52.
libgcc	-
libltdl7	Required when using ODBC drivers.
liblz4-1_7	-
liblzma5	-
libmemcached	Required when using Pgpool-II.
libncurses5	-
libstdc++	-
libz1	-

Package name	Remarks
libzstd1	-
llvm	Version 7.0.x of llvm is required to run SQL with runtime compilation (just-in-time compilation). Install the package that contains libLLVM.so.7. For example, the "libLLVM7" published in SLES12-SP5-Updates includes libLLVM.so.7. Fujitsu Enterprise Postgres uses runtime compilation by default.If you do not want to use runtime compilation, turn off the jit parameter in postgresql.conf.You do not need to install llvm if you turn off the jit parameter. Failure to install llvm without turning off the jit parameter may result in an error when executing SQL.For more information about runtime compilation, see "Just-in-Time Compilation (JIT)" in "PostgreSQL Documentation".
LLVM-libs	Install version 5.0.2 or later.
net-tools	-
pam	Required when using PAM authentication.
perl-libs	Required when using PL/Perl. Install 5.18.
python3	Required when using PL/Python based on Python 3. Install 3.4.x.
rsync	Required when using Pgpool-II.
sysstat	Required when using FJQSS. Set up the sar command after installation.
tcl	Required when using PL/Tcl. Install 8.6.
unzip	-

- The following packages are required for operations on SLES 15.

Package name	Remarks
dstat	Required when using parallel scan.
gdb	-
glibc	-
glibc-32bit	-
iputils	Required for Mirroring Controller.
JRE 8	Required when using the installer, database multiplexing, and WebAdmin. The following JREs are available: - Oracle JRE Use update 31 or later of the 64-bit version. - OpenJDK Past the Java TCK (Technology Compatibility Kit)
libasound2	-
libaudit1	-
libfreebl3	-

Package name	Remarks
libicu	Provides collation support. Install 65.
libgcc	-
libltdl7	Required when using ODBC drivers.
liblz4-1	-
liblzma5	-
libmemcached	Required when using Pgpool-II.
libncurses5	-
libstdc++	-
libz1	-
libzstd1	-
llvm	Version 7.0.x of llvm is required to run SQL with runtime compilation (just-in-time compilation). Install the package that contains libLLVM.so.7. For example, the "libLLVM7" published in SLE-Module-Basesystem15-SP3-Updates includes libLLVM.so.7. Fujitsu Enterprise Postgres uses runtime compilation by default.If you do not want to use runtime compilation, turn off the jit parameter in postgresql.conf.You do not need to install llvm if you turn off the jit parameter. Failure to install llvm without turning off the jit parameter may result in an error when executing SQL.For more information about runtime compilation, see "Just-in-Time Compilation (JIT)" in "PostgreSQL Documentation".
LLVM-libs	Install version 5.0.2 or later.
net-tools	-
pam	Required when using PAM authentication.
perl	Required when using PL/Perl. Install 5.26.
protobuf-c	Required if using the Transparent Data Encryption feature when using a key management system as a keystore. Install 1.3.2.
python3	Required when using PL/Python based on Python 3. Install 3.6.x.
rsync	Required when using Pgpool-II.
sysstat	Set up the sar command after installation.
tcl	Required when using PL/Tcl. Install 8.6.
unzip	-

2.2 Related Software

The following table lists the software required to use Fujitsu Enterprise Postgres.

Table 2.1 Related software

No.	Product name	Version	Fujitsu Enterprise Postgres product name		Remarks
			AE	SE	
1	PRIMECLUSTER (*1)	4.5A10 (*1)	Y	Y	Mandatory when performing failover operation integrated with PRIMECLUSTER.

*1: The applicable products are shown below:

- PRIMECLUSTER Enterprise Edition
- PRIMECLUSTER HA Server

The following table lists servers that can be connected to the Fujitsu Enterprise Postgres client feature.

The following table lists client that can be connected to the Fujitsu Enterprise Postgres server feature.

Table 2.2 Connectable client

OS	Product name
Windows Linux	Fujitsu Enterprise Postgres Client 15 or later

Note

The connection from a client product of a different version to this server function depends on the compatibility of each function included in the client product with PostgreSQL, so some functions may not be available.

The following table lists server assistant that can be connected to the Fujitsu Enterprise Postgres server feature.

Table 2.3 Connectable server assistant

OS	Product name
Linux Windows	Fujitsu Enterprise Postgres Advanced Edition 15

2.3 Excluded Software

This section describes excluded software.

Fujitsu Enterprise Postgres

Fujitsu Enterprise Postgres cannot be installed if all the following conditions are met:

- The product version (indicated by "*x*" in "*x* SP*z*") of the product to be installed is the same as that of the installed product
- The editions are different

Example

In the following cases, Fujitsu Enterprise Postgres cannot be installed as an exclusive product:

- The installed product is Fujitsu Software Enterprise Postgres Standard Edition (64bit) 15
- The product to be installed is Fujitsu Software Enterprise Postgres Advanced Edition (64bit) 15

Other products

There are no exclusive products.

2.4 Required Patches

There are no required patches.

2.5 Hardware Environment

The following hardware is required to use Fujitsu Enterprise Postgres.

Memory

At least 512 MB of memory is required.

2.6 Disk Space Required for Installation

The following table shows the disk space requirements for new installation of Fujitsu Enterprise Postgres. If necessary, increase the size of the file system.

RHEL

Directory	Required disk space (Unit: MB)
/etc	1 + 1 (*1)
/var	1 + 2 (*1) +1 (*2)
/opt	283 (*1) +1 (*2)
Installation destination of the server	730
Installation destination of WebAdmin	730
Installation destination of the client (64-bit)	160
Installation destination of Pgpool-II	35

*1: Required for the installation of the Uninstall (middleware) tool.

*2: Required for the installation of FJQSS.

SLES

Directory	Required disk space (Unit: MB)
/etc	1 + 1 (*1)
/var	1 + 2 (*1) +1 (*2)
/opt	2 (*1) +1 (*2)
Installation destination of the server	230
Installation destination of WebAdmin	390
Installation destination of the client (64-bit)	150
Installation destination of Pgpool-II	35

*1: Required for the installation of the Uninstall (middleware) tool.

*2: Required for the installation of FJQSS.

2.7 Supported System Environment

This section describes the supported system environment.

2.7.1 TCP/IP Protocol

Fujitsu Enterprise Postgres supports version 4 and 6 (IPv4 and IPv6) of TCP/IP protocols.

Note

Do not use link-local addresses if TCP/IP protocol version 6 addresses are used.

2.7.2 File System

All file systems with a POSIX-compliant interface are supported.

However, for stable system operation, the disk where the database is deployed must use a highly reliable file system. Consider this aspect when selecting the file system to be used.

The recommended file system is "ext4".

2.8 PostgreSQL Version Used for Fujitsu Enterprise Postgres

Fujitsu Enterprise Postgres is based on PostgreSQL 15.0.

2.9 Notes on Using Streaming Replication

To use streaming replication, build the primary server and all standby servers using the same Fujitsu Enterprise Postgres version (*1).

*1: The product version is indicated by "*x*" in the notation "*x* SP*z*".

Note

Streaming replication cannot be used in combination with Open Source PostgreSQL.

2.10 Key Management System Requirements

Describes the requirements for a key management system.

2.10.1 To Connect to a key Management System Using the KMIP Protocol

If you use a key management system as a keystore to use the Transparent Data Encryption feature, the following conditions must be met.

Protocol

Key management systems must use the Key Management Interoperability Protocol (KMIP) Version 1.4 protocol.

Encryption Key

The encryption key used must be able to be created or brought into the KMIP server under the following conditions.

- AES 256 bit symmetric key

  A Managed Object that meets the following criteria:

  - Cryptographic Algorithm : AES
  - Cryptographic Length : 256

- Key not wrapped

Operation

The following operations using the KMIP protocol must be supported:

- Get operation

  Encryption keys can be returned in Key Format Type: Raw format.

Client authentication

You must be able to authenticate and authorize clients in the following ways:

- The registered client certificate can authenticate the client and authorize access to the encryption key.

Quantitative Limits

Fujitsu Enterprise Postgres can receive a maximum response size of 8192 bytes from a key management system. Any further response results in an error.

If the private key file used for the client certificate is encrypted, the maximum length of the passphrase used for encryption is 1023 bytes.

2.10.2 To Connect to a Key Management System Using a Plug-in

If you are using a key management system that requires a connection using a protocol other than KMIP, you will need an adapter that converts the request from the Fujitsu Enterprise Postgres into a request format that the key management system can accept.

By preparing the adapter and registering it as a plug-in to the Fujitsu Enterprise Postgres, you can use the key management system as a keystore.

The adapter must be implemented to meet the requirements specified by the Fujitsu Enterprise Postgres.

The key management system must be capable of meeting the requirements of the adapter.

See "Appendix M Access to Key Management System Using Plug-in" for adapter requirements.

Quantitative Limits

The maximum length of the secret, used for example to pass passwords and other information to plug-ins, is 4095 bytes.

Chapter 3 Installation

This chapter explains each of the installation procedures of Fujitsu Enterprise Postgres.

3.1 Pre-installation Tasks

Check the system environment below before installing Fujitsu Enterprise Postgres.

Check the disk space

Ensure that there is sufficient disk space to install Fujitsu Enterprise Postgres.

Refer to "2.6 Disk Space Required for Installation" for information on the required disk space.

Reconfigure the disk partition if disk space is insufficient.

Set JAVA_HOME (SLES only)

Ensure that JRE 8 is installed, and export the JAVA_HOME environment variable.

```
#export JAVA_HOME="Jre8InstallDir"
```

Refer to "Appendix L Procedure when Modifying the JRE Installation" for information on modifying JRE after installation.

Executable Users

Installation and uninstallation is performed by one of the following users.

- superuser

  On the system, run the following command to become superuser.

```
$ su -
Password:******
```

- System Administrator

  Check the OS documentation for system administrators.

Note

When you execute a command as a system administrator, there are multiple commands that require superuser privileges.

Check the OS documentation before setting. Specify all in the command field.

Check the installed products and determine the installation method

Using the operation shown below, start Uninstall (middleware), and check the installed products.

Example

```
# /opt/FJSVcir/cimanager.sh -c
Loading Uninstaller...


Currently installed products
1. productName
2. productName
3. productName
4. productName
5: productName

Type [number] to select the software you want to uninstall.
```

```
[number,q]
=>q



Exiting Uninstaller.
```

If Fujitsu Enterprise Postgres is already installed, determine which installation method to use:

- Reinstallation
- Multi-version installation

Remove applied updates

If you perform reinstallation as the installation method, remove applied updates using the procedure shown below.

Note

If a product is installed without removing applied updates, the following problems will occur:

- Performing reinstallation

  If an update with the same update and version number is applied, an error informing you that the update has already been applied is displayed.
  Perform reinstallation after removing the update.

1. Display the applied updates

   Execute the following command to display the applied updates:

```
# /opt/FJSVfupde/bin/uam showup
```

2. Remove the updates

   Execute the command below to remove the updates. If an update with the same update number was applied more than once, the updates are removed in order, starting from the highest version number.

```
# /opt/FJSVfupde/bin/uam remove -i update-number
```

   Note

   If the installation directory/lib is set in the environment variable LD_LIBRARY_PATH for the running user, remove the installation directory/lib from LD_LIBRARY_PATH.

Determine the preferred WebAdmin configuration

Starting with Fujitsu Enterprise Postgres 9.5, WebAdmin can be installed in two configurations:

- Single-server
- Multiserver

See

Refer to "Appendix I Determining the Preferred WebAdmin Configuration" for details.

Determining the Pgpool-II System Configuration

The system configuration when using Pgpool-II is as follows:

- Place on database server
- Place on application server
- Place on dedicated server

See

Refer to "Appendix J System Configuration when using Pgpool-II".

3.2 Installation in Interactive Mode

Install according to the following procedure:

Note

- The following characters can be used as input values:

  Alphanumeric characters, hyphens, commas and forward slashes

- When reinstalling the product, back up the following folder in which the WebAdmin instance management information is stored:

  webAdminInstallFolder/data/fepwa

  Follow the procedure below to perform the backup.

  1. Stop the WebAdmin server. Refer to "B.1.3 Stopping the Web Server Feature of WebAdmin" for details.
  2. Back up the following folder:

     webAdminInstallFolder/data/fepwa

  Replace the above folder with the backed up folder when the reinstallation is complete.

Point

For installation in interactive mode, default values are set for the installation information. The following settings can be changed for a new installation or a multi-version installation:

- Installation directory
- WebAdmin setup information, if WebAdmin is selected

  To change the port number, confirm that it is an unused port number between 1024 and 32767.

1. Stop applications and programs

If the installation method is the following, all applications and programs that use the product must be stopped:

- Reinstallation

Before starting the installation, stop the following:

- Applications that use the product
- Connection Manager
- Instance
- Web server feature of WebAdmin

  Execute the WebAdminStop command to stop the Web server feature of WebAdmin.

Example

If WebAdmin is installed in /opt/fsepv<*x*>webadmin:

```
# cd /opt/fsepv<x>webadmin/sbin
# ./WebAdminStop
```

- Mirroring Controller

  Execute the mc_ctl command with the stop mode option specified and stop the Mirroring Controller.

  Example

  ```
  $ mc_ctl stop -M /mcdir/inst1
  ```

- pgBadger
- Pgpool-II

2. Mount the DVD drive

Insert the server program DVD into the DVD drive, and run the command given below.

Example

```
# mount -t iso9660 -r -o loop /dev/dvd /media/dvd
```

Here /dev/dvd is the device name for the DVD drive (which may vary depending on your environment), and /media/dvd is the mount point (which may need to be created before calling the command).

Note

If the DVD was mounted automatically using the automatic mount daemon (autofs), "noexec" is set as the mount option, so the installer may fail to start. In this case, use the mount command to remount the DVD correctly, and then run the installation. Note that the mount options of a mounted DVD can be checked by executing the mount command without any arguments.

3. Run the installation

Execute the following command:

Example

```
# cd /media/dvd
# ./install.sh
```

In the example above, /media/dvd is the DVD mount point.

4. Select the product for installation

The list of installation target products is displayed.
Type the number for the product to be installed, or "all", and press Enter.

Information

- The Fujitsu Enterprise Postgres server component and WebAdmin can be installed on the same machine by selecting the "Fujitsu Enterprise Postgres server component" and the "WebAdmin component".
- Pgpool-II to be installed on the same server as the database server, you can install it on the same machine by selecting "Fujitsu Enterprise Postgres Server Component" and "Pgpool-II component". Pgpool-II to be installed on the same server as the application server, or on a dedicated server that is different from the database server and application server, see "Appendix J System Configuration when using Pgpool-II".

- If the selected product has been installed, a window for selecting reinstallation or multi-version installation is displayed for each product. Follow the on-screen instructions to select the installation method.

5. Check the settings

The window for checking the installation information is displayed.
Type "y" and press Enter to start the installation.
To change the settings, type "c", press Enter, and follow the on-screen instructions. This option is not displayed if there is no information that can be modified.

If you have not set up WebAdmin during installation, refer to "Appendix B Setting Up and Removing WebAdmin" for details.

6. Check the changed settings

If the installation information has been modified, the new installation information is displayed.
Type "y" and press Enter to start the installation.
To change the settings again, type "c" and press Enter.

7. Display the installation status

When the installation is started, the installation status is displayed.

Note that even if WebAdmin setup fails, the process will continue.

If the following message is displayed, manually execute WebAdmin setup after installation is completed.
Refer to "B.1 Setting Up WebAdmin" for information on the WebAdmin setup procedure.

```
Starting setup.

Setting up WebAdmin.

ERROR: WebAdmin setup failed.
```

8. Finish the installation

When the installation is complete, a message is displayed showing the status.
If installation was successful, a message like the one shown below is displayed:

```
Installed successfully.
```

Note

If an error occurs during the installation, read the error message and remove the cause of the error, and then reexecute the install.sh command.

3.3 Installation in Silent Mode

Installation in silent mode can be performed only when the installation method is one of the following:

- New installation
- Multi-version installation

See

Refer to the Fujitsu Enterprise Postgres product website for information on installation in silent mode, such as the installation parameters and error messages.

The installation procedure is described below.

1. Mount the DVD drive

Insert the server program DVD into the DVD drive, and run the command given below.

Example

```
# mount -t iso9660 -r -o loop /dev/dvd /media/dvd
```

Here /dev/dvd is the device name for the DVD drive (which may vary depending on your environment), and /media/dvd is the mount point (which may need to be created before calling the command).

Note

If the DVD was mounted automatically using the automatic mount daemon (autofs), "noexec" is set as the mount option, so the installer may fail to start. In this case, use the mount command to remount the DVD correctly, and then run the installation. Note that the mount options of a mounted DVD can be checked by executing the mount command without any arguments.

2. Create an installation parameters CSV file

Consider the features that will be required for system operations, and then create an installation parameters CSV file that uses the following specification format.

```
sectionName, parameterName, value
sectionName, parameterName, value
    :
```

Information

The template for the installation parameters CSV file is "*mountPoint*/sample/sample.csv".

3. Run the installation

Execute the following command:

Example

```
# cd /media/dvd
# ./silent.sh /home/work/inspara.csv
```

In the example above, /media/dvd is the DVD mount point, and /home/work/inspara.csv is the installation parameter CSV.

If the installer ends in an error, a message is output to the log file and return values are returned.

Chapter 4 Setup

This chapter describes the setup procedures to be performed after installation completes.

4.1 Operating Method Types and Selection

This section describes how to operate Fujitsu Enterprise Postgres.

There are two methods of managing Fujitsu Enterprise Postgres operations - select one that suits your purposes:

The Operation Guide describes the operating method using WebAdmin, and the equivalent operating method using the server commands.

Simple operation management using a web-based GUI tool (WebAdmin)

Suitable when using frequently used basic settings and operations for operation management.

This method allows you to perform simple daily tasks such as starting the system before beginning business, and stopping the system when business is over, using an intuitive operation.

Usage method

Usage is started by using WebAdmin to create the instance.

By using an external scheduler and the pgx_dmpall command, periodic backups can be performed, which can then be used in recovery using WebAdmin.

Note

Do not use a server command other than pgx_dmpall and pgx_keystore or a server application. Operation modes that use server commands and server applications cannot be used in conjunction with WebAdmin. If used, WebAdmin will not be able to manage the instances correctly.
In addition, to perform a backup by copy command from the pgx_dmpall command, select the operating method using the server commands.
Refer to Reference and the PostgreSQL Documentation for information on server commands and server applications.

Advanced operation management using server commands

When operating in a system that is automated by operation management middleware (Systemwalker Centric Manager, for example), this method allows you to use more detailed settings and operations and perform higher level operation management.

An overview of the operating method using the GUI, and its relationship with the operating method using the server commands, are shown below.

Refer to the Operation Guide for details.

Operation		Operation with the GUI	Operation with commands
Setup	Creating an instance	WebAdmin is used. The server machine capacity, and the optimum parameter for operations using WebAdmin, are set automatically.	The configuration file is edited directly using the initdb command.
	Creating a standby instance	WebAdmin is used. WebAdmin performs a base backup of the source instance and creates a standby instance.	A standby instance is created using the pg_basebackup command.
	Changing the configuration files	WebAdmin is used.	The configuration file is edited directly.
Starting and stopping an instance		WebAdmin is used.	The pg_ctl command is used.

Operation		Operation with the GUI	Operation with commands
Creating a database		This is defined using pgAdmin of the GUI tool, or using the psql command or the application after specifying the DDL statement.	
Backing up the database		WebAdmin, or the pgx_dmpall command, is used.	It is recommended that the pgx_dmpall command be used. Recovery to the latest database can be performed.
Database recovery		WebAdmin is used.	To use the backup that was performed using the pgx_dmpall command, the pgx_rcvall command is used.
Monitoring	Database errors	The status in the WebAdmin window can be checked. (*1)	The messages that are output to the database server log are monitored (*1)
	Disk space	The status in the WebAdmin window can be checked. A warning will be displayed if the free space falls below 20%. (*1)	This is monitored using the df command of the operating system, for example. (*1)
	Connection status	This can be checked using pgAdmin of the GUI tool, or referencing pg_stat_activity of the standard statistics view from psql or the application.	

*1: This can be used together with system log monitoring using operations management middleware (Systemwalker Centric Manager, for example).

See

Refer to "Periodic Operations" and "Actions when an Error Occurs" in the Operation Guide for information on monitoring and database recovery.

4.2 Preparations for Setup

This section describes the preparation required before setting up Fujitsu Enterprise Postgres.

4.2.1 Creating an Instance Administrator

Decide which OS user account will be assigned the instance administrator role. You can assign it to a new user or to an existing one, but you cannot assign it to the OS superuser (root).

The following example shows an OS user account with the name "fsepuser" being assigned the instance administrator role.

Example

```
# useradd fsepuser
# passwd fsepuser
```

Note

The following note applies if using WebAdmin for operations:

- If the password is changed for the user account of the instance administrator, set the changed password using ALTER ROLE WITH ENCRYPTED PASSWORD.

4.2.2 Preparing Directories for Resource Deployment

Prepare the directories required when creating instances.

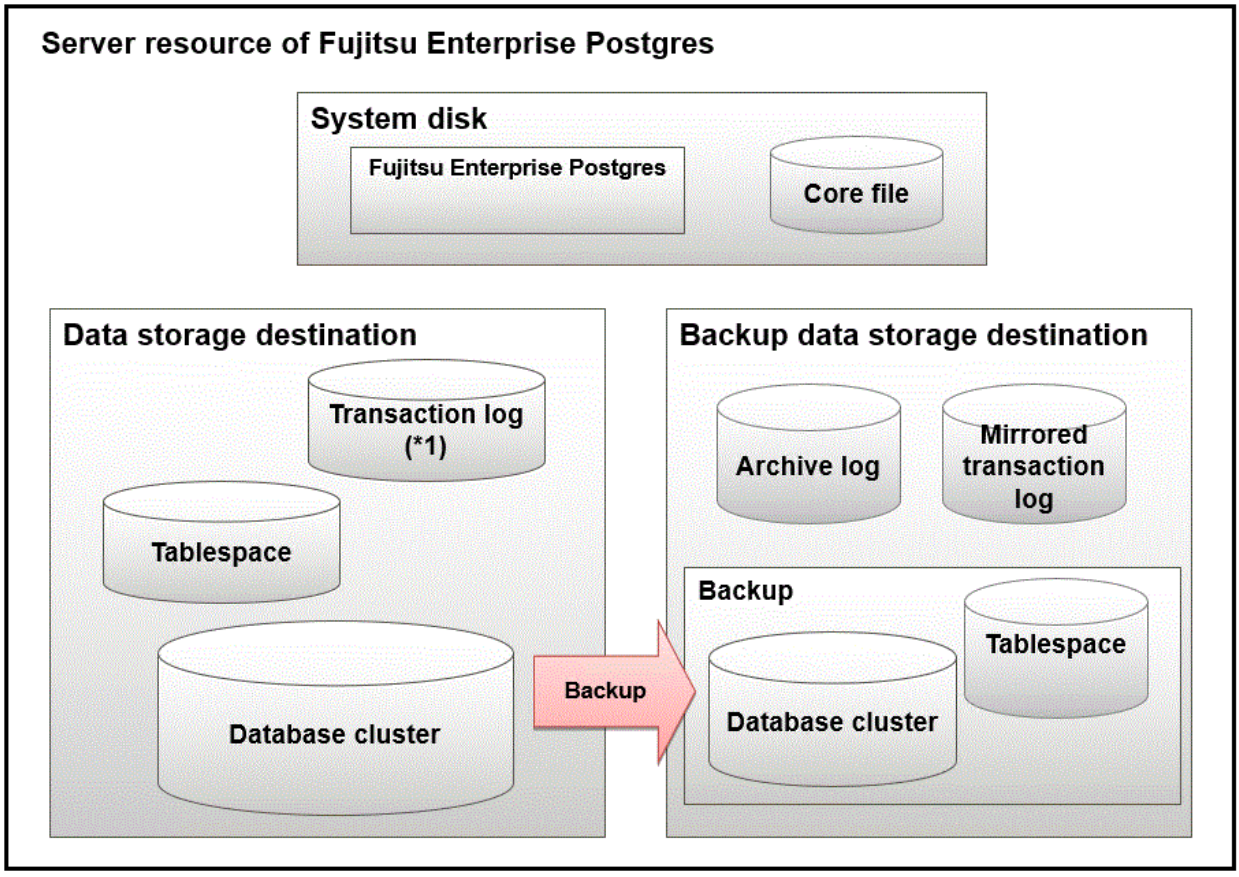
Considerations when deploying resources

The disk configuration on the resource deployment destination is important, because it affects not only recovery following disk corruption, but normal operation as well. The points for determining the disk configuration are as follows:

1. If the backup data storage destination and the data storage destination are both lost, it will not be possible to recover the data, so deploy them to separate disks.
2. To shorten the recovery time following a single disk fault, deploy the system disk and data storage destination to separate disks.
3. The backup data storage destination requires at least double the capacity of the data storage destination, so deploy it to the disk with the most space available.
4. When large amounts of data are updated, the write-to load for the data storage destination, transaction log storage destination, and backup data storage destination (mirrored transaction log) will also be great. For this reason, deploy them to separate disks, out of consideration for performance.

Note

When using the volume manager provided by the operating system, be aware of which physical disk the file system has been created on, for example, by deploying the data storage destination and the backup data storage destination to separate disks.



*1: To distribute the I/O load, place the transaction log on a different disk from the data storage destination.

Resource	Role
Database cluster	The area where the database is stored. It is a collection of databases managed by an instance.
Tablespace	Stores table files and index files in a separate area from the database cluster. Specify a space other than that under the database cluster.
Transaction log	Stores log information in preparation for a crash recovery or rollback. This is the same as the WAL (Write Ahead Log).
Archive log	Stores log information for recovery
Mirrored transaction log (mirrored WAL)	Enables a database cluster to be restored to the state immediately before an error even if both the database cluster and transaction log fail when performing backup/recovery operations using the pgx_dmpall command or WebAdmin.
Corefile	Fujitsu Enterprise Postgres process corefile output when an error occurs with a Fujitsu Enterprise Postgres process.

Examples of disk deployment

The following are examples of disk deployment:

Number of disks	Disk	Deployment
3	System disk	Fujitsu Enterprise Postgres program
		Corefile
	Connected physical disk	Data storage destination, transaction log storage destination
	Connected physical disk	Backup data storage destination
2	System disk	Fujitsu Enterprise Postgres program
		Corefile
		Data storage destination, transaction log storage destination
	Connected physical disk	Backup data storage destination

Proposal for disk deployment using WebAdmin

To generate an instance using WebAdmin, we recommend an optimum deployment that takes into account the status of all disks at the time of instance generation, and items 1 to 3 in the "Considerations when deploying resources" subheading above, based on the limitations below (note that a different deployment can also be specified).

- The mount point does not include national characters
- The instance administrator has the proper permissions to read and write on the mount point

Preparing directories

The directories to be prepared depend on the way that you create the instances.

The following shows the directories that need to be prepared:

Directory to be prepared	Using WebAdmin	Using the initdb command
Data storage destination	Y (*1)	Y
Backup data storage destination	O (*1) (*4)	O

Directory to be prepared	Using WebAdmin	Using the initdb command
Transaction log storage destination	O (*1) (*2)	O
Corefile output destination	N (*3)	O

Y: Required
O: Optional
N: Not required

*1: WebAdmin automatically creates a directory

*2: The default is to create in a directory in the data storage destination. When it is necessary to distribute the I/O load for the database data and the transaction log, consider putting the transaction log storage destination on a different disk from the data storage destination

*3: The corefile path is as follows:

`/var/tmp/fsep_`*`version`*`/`*`instanceAdminUser`*`_`*`instanceNamePortNumber`*`/core`

version: product version_WA_architecture

Note: The product version is normally the version of WebAdmin used to create the instance. For example, WebAdmin 15 allows a user to create a FUJITSU Enterprise Postgres 9.6 instance on a database server having WebAdmin 9.6. In this case, because WebAdmin 9.6 is used to create the instance, the product version will be "96".

instanceAdminUser: operating system user name

PortNumber: port number specified when creating the instance

Example:

/var/tmp/fsep_150_WA_64/naomi_myinst27599/core

Note that resources placed in /var/tmp that have not been accessed for 30 days or more will be deleted by the default settings of the operating system. Consider excluding them from deletion targets or changing the output destination in the operating system settings.

To change the output destination, configure the core_directory and the core_contents parameters in postgresql.conf. Refer to "Parameters" in the Operation Guide for information on the settings for these parameters.

*4: This directory is required when instance backup is enabled.

Note

- The directories must meet the following conditions:
  - The directory owner must be the OS user account that you want to be the instance administrator
  - The directory must have write permission
  - The directory must be empty
- If you use WebAdmin, you cannot use directories mounted over the network.
  Examples include NFS (Network File System) and CIFS (Common Internet File System).
  Also, even if you are not using WebAdmin, do not use these directories unless you are creating tablespaces on a storage device on your network.

Example

The following example shows the OS superuser creating /database/inst1 as the directory for storing the database data and changing the owner of the directory to the OS user account "fsepuser".

```
# mkdir -p /database/inst1
# chown -R fsepuser:fsepuser /database/inst1
# chmod 700 /database/inst1
```

4.2.3 Estimating Resources

Estimate the resources to be used on the Fujitsu Enterprise Postgres.

Refer to "Appendix E Estimating Database Disk Space Requirements" for information on estimating database disk space requirements.

Refer to "Parameters automatically set by WebAdmin according to the amount of memory" when creating multiple instances with WebAdmin.

Refer to "Appendix F Estimating Memory Requirements" when creating instances with the initdb command, to estimate memory usage.

4.2.4 Configuring Corefile Names

If a process crashes, a corefile for the process will be generated by the operating system. If a corefile is generated with the same name as an existing corefile generated for a different process, the newly-generated corefile will overwrite the previously dumped corefile. To prevent this, configure a unique corefile name for each crash by appending the process ID, program name, and datetime.

Corefile names can be configured using the "kernel.core_pattern" and "kernel.core_uses_pid" kernel parameters.
Refer to the "man page" in "core(5)" for information on how to use these parameters.

Note that with regard to the location for storing corefiles, the operating system settings take precedence over the core_directory parameter of postgresql.conf.

If you specify systemd-coredump as the core_pattern, the core file is not placed in the location specified by the core_directory parameter.See the systemd-coredump (8) man page for the location of core files.

Use coredumpctl to retrieve core files.For more information about using coredumpctl, see the coredumpctl (1) man page.

4.3 Creating Instances

There are two methods that can be used to create an instance:

- 4.3.1 Using WebAdmin
- 4.3.2 Using the initdb Command

Creating multiple instances

Multiple instances can be created.

The memory allocated needs to be adjusted when multiple instances are created with WebAdmin (refer to "Parameters automatically set by WebAdmin according to the amount of memory" for details).

Features that cannot be set up using WebAdmin

The "Storage data protection using transparent data encryption" feature cannot be set up using WebAdmin.

To set up this feature in an instance created by WebAdmin, perform the additional setup tasks detailed in "Storage Data Protection using Transparent Data Encryption" in the Operation Guide.

Note

- Instances created using the initdb command (command line instances) can be managed using WebAdmin, however, they must first be imported into WebAdmin. Refer to "4.3.1.4 Importing Instances" for details.
- Always use WebAdmin to delete instances that were created or imported using WebAdmin. Because WebAdmin management information cannot be deleted, WebAdmin will determine that the instance is abnormal.
- Databases with the names 'template0' and 'template1' are automatically created when an instance is created.
  These databases are used as the templates for databases created later. Furthermore, a default database with the name 'postgres' is automatically created, which will be used with Fujitsu Enterprise Postgres commands. It is important that you do not delete these databases created by default.

4.3.1 Using WebAdmin

This section describes how to create an instance using WebAdmin.

WebAdmin must be set up correctly before it can be used. Refer to "B.1 Setting Up WebAdmin" for details. Additionally, if WebAdmin needs to be configured to use an external repository database, refer to "B.3 Using an External Repository for WebAdmin" for details.

It is recommended to use the following browsers with WebAdmin:

- Microsoft Edge (Build41 or later)

WebAdmin will work with other browsers, such as Firefox and Chrome, however, the look and feel may be slightly different.

Configure your browser to allow cookies and pop-up requests from the server on which Fujitsu Enterprise Postgres is installed.

Refer to "Appendix A Recommended WebAdmin Environments" for information on how to change the pop-up request settings and other recommended settings.

- WebAdmin does not run in Windows(R) safe mode.
- If the same instance is operated from multiple WebAdmin windows, it will not work correctly.
- If the same instance is operated from multiple WebAdmin versions, it will not work correctly. Always use the latest version of WebAdmin for instance operations.
- For efficient use of WebAdmin, it is recommended not to use the browser [Back] and [Forward] navigation buttons, the [Refresh] button, and context-sensitive menus, including equivalent keyboard shortcuts.
- Copying and pasting the WebAdmin URLs are not supported. Additionally, bookmarking of WebAdmin URLs is not supported.
- It is recommended to match the language between the instance server locale and WebAdmin.
- WebAdmin supports only two languages: English and Japanese.
- It is recommended to change the WebAdmin language setting from the instance details page only.
- It is recommended to operate WebAdmin using the WebAdmin launcher.
- WebAdmin uses the labels "Data storage path", "Backup storage path" and "Transaction log path" to indicate "data storage destination", "backup data storage destination" and "transaction log storage destination" respectively. In this manual these terms are used interchangeably.
- If the browser was not operated for a fixed period (about 30 minutes), the session will time out and the login page will be displayed again for the next operation.
- Port access permissions
  If a port is blocked (access permissions have not been granted) by a firewall, enable use of the port by granting access. Refer to the vendor document for information on how to grant port access permissions.
  Consider the security risks carefully when opening ports.
- When creating or importing an instance in WebAdmin, set the log_directory parameter in postgresql.conf to '/var/tmp/fsep_*version*/*instanceAdminUser_instanceNamePortNumber*/log'. Note that resources placed in /var/tmp that have not been accessed for 30 days or more will be deleted by the default settings of the operating system. Therefore, consider excluding instances created using WebAdmin from deletion targets in the operating system settings if you need to stop those instances for a long time.

4.3.1.1 Logging in to WebAdmin

This section describes how to log in to WebAdmin.

Startup URL for WebAdmin

In the browser address bar, type the startup URL of the WebAdmin window in the following format:

```
http://hostNameOrIpAddress:portNumber/
```

- *hostNameOrIpAddress*: Host name or IP address of the server where WebAdmin is installed.
- *portNumber*: Port number of WebAdmin. The default port number is 27515.

The startup screen is displayed. From this window you can log in to WebAdmin or access the product documentation.

Logging in to WebAdmin

Click [Launch WebAdmin] in the startup URL window to start WebAdmin and display the login window. Enter the instance administrator user ID (operating system user account name) and password, and log in to WebAdmin. User credential (instance administrator user ID and password) should not contain hazardous characters. Refer to "Appendix C WebAdmin Disallow User Inputs Containing Hazardous Characters".

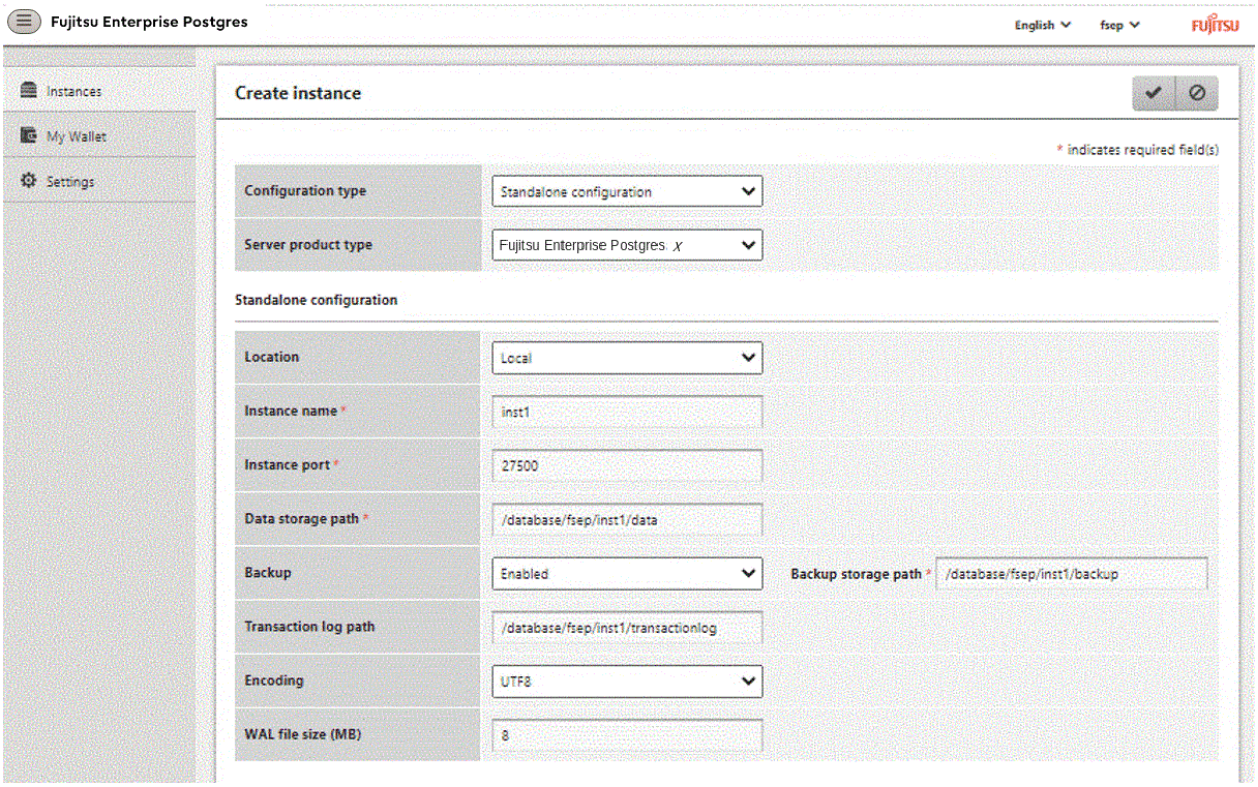
4.3.1.2 Creating an Instance

This section describes how to create an instance.

Information

WebAdmin calculates values using the formula indicated in "Managing Kernel Resources" under "Server Administration" in the PostgreSQL Documentation, and configures these in the kernel parameters. Refer to "Appendix H Configuring Kernel Parameters" for information on configuring parameter names.
Refer to "Appendix D Configuring Parameters" for information on the postgresql.conf values required to derive the set values.

1. Start WebAdmin, and log in to the database server.
2. In the [Instances] tab, click ➕.
3. Enter the information for the instance to be created.



Enter the following items:

- [Configuration type]: Whether to create a standalone instance or an instance that is part of a cluster.

- [Server product type]: Sets which of the following instances to create:
  - Fujitsu Enterprise Postgres 9.5 Instances
  - Fujitsu Enterprise Postgres 9.6 Instances
  - Fujitsu Enterprise Postgres 10 Instances
  - Fujitsu Enterprise Postgres 11 Instances
  - Fujitsu Enterprise Postgres 12 Instances
  - Fujitsu Enterprise Postgres 13 Instances
  - Fujitsu Enterprise Postgres 14 Instances
  - Fujitsu Enterprise Postgres 15 Instances

  The default is "Fujitsu Enterprise Postgres 15".
  WebAdmin can create and manage instances compatible with the following, but new functionality in Fujitsu Enterprise Postgres 15 may not support the instance or it may be disabled.
  - Fujitsu Enterprise Postgres 9.5
  - Fujitsu Enterprise Postgres 9.6
  - Fujitsu Enterprise Postgres 10
  - Fujitsu Enterprise Postgres 11
  - Fujitsu Enterprise Postgres 12
  - Fujitsu Enterprise Postgres 13
  - Fujitsu Enterprise Postgres 14
- [Location]: Whether to create the instance in the server that the current user is logged into, or in a remote server. The default is "Local", which will create the instance in the server machine where WebAdmin is currently running.
- [Instance name]: Name of the database instance to manage

  The name must meet the conditions below:
  - Maximum of 16 characters
  - The first character must be an ASCII alphabetic character
  - The other characters must be ASCII alphanumeric characters
- [Instance port]: Port number of the database server
- [Data storage path]: Directory where the database data will be stored
- [Backup]: Whether to enable or disable the WebAdmin backup feature. The default is "Enabled". Select "Disabled" to disable all backup and restore functionality for the instance. If "Enabled" is selected, enter the following item:
  - [Backup storage path]: Directory where the database backup will be stored
- [Transaction log path]: Directory where the transaction log will be stored
- [Encoding]: Database encoding system
- [WAL file size]: Allow the WAL file size to be set when creating an instance. The default is 16 MB if the field is blank. The size specified must be a power of 2 between 1 and 1024. This option is not available for standby instances.

If "Remote" is selected for [Location], enter the following additional items:

- [Host name]: Name of the host where the instance is to be created
- [Operating system credential]: Operating system user name and password for the remote machine where the instance is to be created
- [Remote WebAdmin port for standalone]: Port in which WebAdmin is accessible in the remote machine

Note

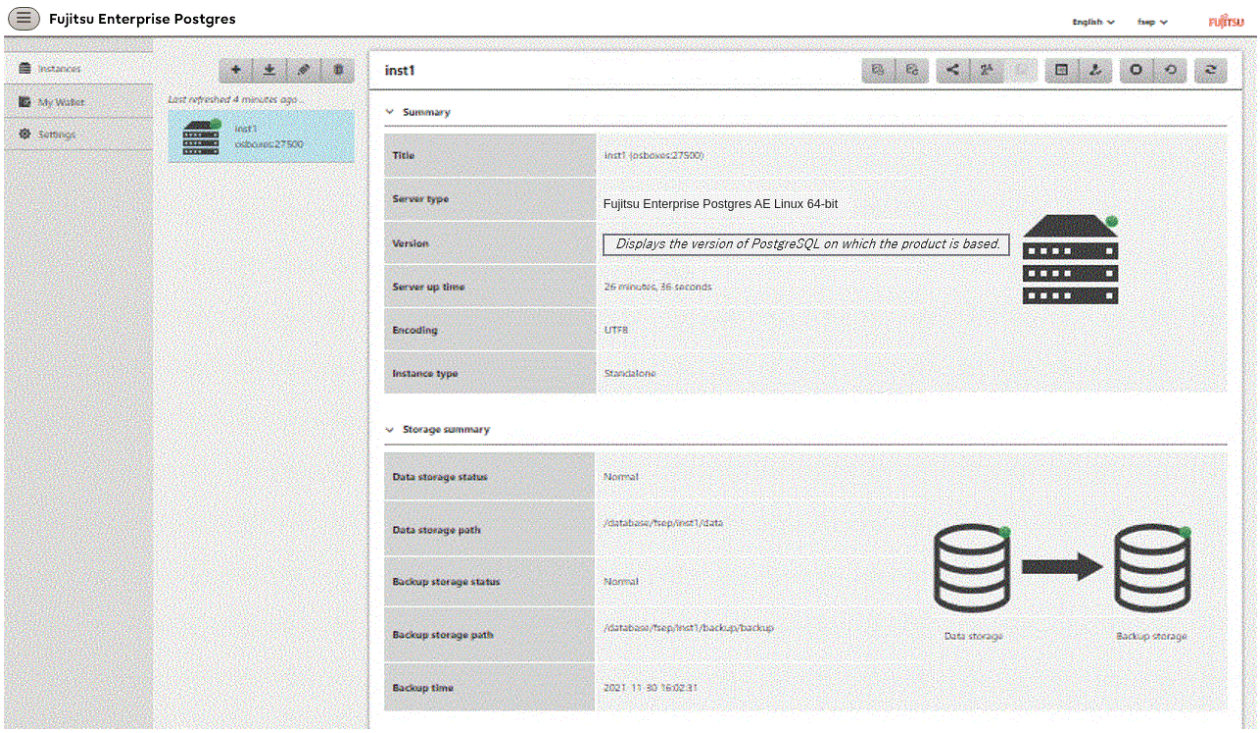
- Refer to "4.2.2 Preparing Directories for Resource Deployment" - "Considerations when deploying resources" for information on points to consider when determining the data storage path, backup storage path, and transaction log path.
- The following items can be modified after the instance has been created. These items cannot be modified on instances that have compatibility with FUJITSU Enterprise Postgres 9.5.
  - Instance name
  - Port number
  - Backup storage path

  Refer to "Editing instance information" for details.
- Do not specify directories that include symbolic link or multibyte characters when specifying the data storage destination or backup data storage destination.
- In the instance that is created using WebAdmin, the locale of the character set to be used in the database, and the locale of the collating sequence, are fixed using C.
- For enhanced security, WebAdmin encrypts the superuser password using SCRAM-SHA-256 authentication for all Enterprise Postgres 10 or later instances. The client/driver must therefore support SCRAM-SHA-256 authentication if they need to connect to FUJITSU Enterprise Postgres 10 or later instances created by WebAdmin with superuser credentials.
- Host name and Operating system credential (Operating system user name and password) should not contain hazardous characters. Refer to "Appendix C WebAdmin Disallow User Inputs Containing Hazardous Characters".

4. Click ✔ to create the instance.

   If the instance is created successfully, a message indicating the same will be displayed.

5. The instance will be started when it is created successfully.



6. Back up the basic information that was set

   Back up the WebAdmin management information periodically to ensure operational continuity when a fault occurs on the system disk. Follow the procedure below to perform the backup.

a. Stop the WebAdmin server. Refer to "B.1.3 Stopping the Web Server Feature of WebAdmin" for details.

b. Back up the following directory:

   webAdminInstallDir/data/fepwa

4.3.1.3 Changing Instance Settings

You can change the information that is set when an instance is created.

Change the following settings to suit the operating and management environment for Fujitsu Enterprise Postgres.

- Instance configuration
  - Character encoding
  - Communication
  - SQL options
  - Memory
  - Streaming replication
- Changing client authentication information
- Editing instance information

Information

These settings are the same as the parameters that can be set in the files shown below. Refer to "Appendix D Configuring Parameters" for information on the equivalence relationship between the item name and the parameter.

- postgresql.conf
- pg_hba.conf

Note

The files shown below can also be modified directly, however if a parameter not described in "Appendix D Configuring Parameters" was edited by mistake, WebAdmin may not run correctly.

- postgresql.conf
- pg_hba.conf

You can also modify the following files directly, but WebAdmin may not work correctly if the records span multiple lines. Therefore, change the record to a single row.

- pg_hba.conf
- pg_ident.conf

Instance configuration

1. Start WebAdmin and log in to the database server.
2. In the [Instances] tab, click .
3. Click to change the configuration.
4. Click to save your changes.

See

Select a client-side encoding system that can be converted to/from the database encoding system. Refer to "Automatic Character Set Conversion Between Server and Client" in "Server Administration" in the PostgreSQL Documentation for information on the encoding system combinations that can be converted.

Changing client authentication information

1. Start WebAdmin and log in to the database server.
2. In the [Instances] tab, click .

   Click to register new authentication information.

   To change authentication information, select the information, and then click .

   To delete authentication information, select the information, and then click .

Note

When creating the instance, do not delete the entry below, because it is a connection required for WebAdmin to monitor the operational status of the database:
Type=local, Database=all, User=all, and Method=md5

Editing instance information

Use the [Edit instance] page to modify the following items for an instance:

- Instance name
- Port number
- Backup storage path

1. In the [Instances] tab, click . The [Edit instance] page is displayed.
2. Modify the relevant items.

   If [Backup storage path] is changed, [Backup management] is enabled. Select the required option:

   Retain existing backup: Create a backup in [Backup storage path] and retain the existing backup in its original location.

   Copy existing backup to new path: Copy the existing backup to [Backup storage path]. A new backup will not be created. The existing backup will be retained in its original location.

   Move existing backup to new path: Move the existing backup to [Backup storage path]. A new backup will not be created.

   Remove existing backup: Create a backup in [Backup storage path]. The existing backup will be removed.
3. Click to save your changes.

Note

- The [Edit instance] page is also displayed when the user selects 'Navigate to the "Edit instance" page' from the [Anomaly Error] dialog box. Refer to "Anomaly Detection and Resolution" in the Operation Guide for information on what takes place when an anomaly is detected.
- When [Instance name] or [Instance port] is modified, the log_directory and core_directory parameters in postgresql.conf are updated. Also, the specified directories are created if they do not exist.

  Refer to "4.3.1.4 Importing Instances" for information on the format of these directories.

4.3.1.4 Importing Instances

Instances can be created using WebAdmin, or via the command line using the initdb command. Instances created using the initdb command (command line instances) can be managed using WebAdmin, however, they must first be imported into WebAdmin.

This section explains how to import command line instances into WebAdmin.

1. In the [Instances] tab, click . The [Import instance] page is displayed.
2. Enter the information for the instance being imported. Refer to "4.3.1.2 Creating an Instance" for information on the items that need to be entered.
3. Click to import the instance.

- Importing neither starts nor stops the instance.
- The following restrictions apply to instance import:
  - Any instance already managed by WebAdmin cannot be imported again.
  - The postgresql.conf file must be located in the same directory as [Data storage path].
  - Read/write permissions are required for [Data storage path].
  - The location specified in postgresql.conf for the following files must not have been changed:
    - hba_file
    - ident_file
  - If the following file contains records that span multiple lines, change the record to a single line before importing.
    - pg_hba.conf
    - pg_ident.conf
  - If the instance is part of a cluster that is monitored by Mirroring Controller, WebAdmin will be unable to detect the Mirroring Controller settings.
  - Instances making use of Mirroring Controller functionality should not be imported, because subsequent operations on those instances may cause unexpected and undesirable side-effects.
  - It is not possible to import and operate an instance that uses a directory mounted by Network File System (NFS).
  - You must make the following changes to the parameters in postgresql.conf prior to importing the instance in WebAdmin.

Parameter	Requirements
port	The port parameter should be uncommented.

The log_directory and core_directory parameters in postgresql.conf are updated during import. Also, the specified directories are created if they do not exist.

The format of these directories is as follows:

log_directory: '/var/tmp/fsep_*version*/*instanceAdminUser*_*instanceNamePortNumber*/log'

core_directory: '/var/tmp/fsep_*version*/*instanceAdminUser*_*instanceNamePortNumber*/core'

version: product version_WA_architecture

instanceAdminUser: operating system user name

PortNumber: port number specified when creating the instance

Examples:

log_directory: '/var/tmp/fsep_150_WA_64/naomi_myinst27599/log'

core_directory: '/var/tmp/fsep_150_WA_64/naomi_myinst27599/core'

- When a standby instance is imported, a valid entry, using the IP address of the standby instance, must exist in the pg_hba.conf file of the master instance to allow the standby instance to connect to the master instance.
- When a standby instance is imported, the value for "host" in the primary_conninfo parameter of postgresql.auto.conf should match the host name of the master instance.
- When a standby instance is imported, you cannot specify "passfile" in the primary_conninfo parameter of postgresql.auto.conf. Be sure to specify "password".
- Instances created by other operating systems cannot be imported.
- If a FUJITSU Enterprise Postgres 10 or later instance is being imported while it is running, WebAdmin will encrypt the superuser password using SCRAM-SHA-256 authentication.

4.3.2 Using the initdb Command

This section describes the procedure to create an instance using the initdb command.

Note

If a port is blocked (access permissions have not been granted) by a firewall, enable use of the port by granting access. Refer to the vendor document for information on how to grant port access permissions.
Consider the security risks carefully when opening ports.

4.3.2.1 Editing Kernel Parameters

Refer to "Appendix H Configuring Kernel Parameters" prior to editing these settings.

After the settings are complete, check the command specifications of the relevant operating system and restart the system if required.

4.3.2.2 Creating an Instance

Create an instance, with the database cluster storage destination specified in the PGDATA environment variable or in the -D option. Furthermore, the user that executed the initdb command becomes the instance administrator.

Note

- Instances created using the initdb command (command line instances) can be managed using WebAdmin, however, they must first be imported into WebAdmin. Refer to "4.3.1.4 Importing Instances" for details.
- If creating multiple instances, ensure that there is no duplication of port numbers or the directories that store database clusters.

See

Refer to "initdb" in "Reference" in the PostgreSQL Documentation for information on the initdb command.

The procedure to create an instance is described below.

1. Use the OS user account that you want as the instance administrator.

   Connect with the server using the OS user account that you want as the instance administrator.

   You cannot use the OS superuser (root).

   The following example shows the OS superuser connected to the server being changed to the OS user account "fsepuser".

Example

```
# su fsepuser
```

2. Configure the environment variables

   Configure the environment variables in the server with the newly created instance.

   Set the following environment variables:

   - PATH environment variables

     Add the installation directory "/bin".

   - MANPATH environment variables

     Add the installation directory "/share/man".

   Example

   The following example configures environment variables when the installation directory is "/opt/fsepv<*x*>server64".

   Note that "<*x*>" indicates the product version.

   sh, bash

   ```
   $ PATH=/opt/fsepv<x>server64/bin:$PATH ; export PATH
   $ MANPATH=/opt/fsepv<x>server64/share/man:$MANPATH ; export MANPATH
   ```

   csh, tcsh

   ```
   $ setenv PATH /opt/fsepv<x>server64/bin:$PATH
   $ setenv MANPATH /opt/fsepv<x>server64/share/man:$MANPATH
   ```

3. Create a database cluster

   Create the database cluster with the initdb command, specifying the storage destination directory.

   Specify the transaction log storage destination and the locale setting option as required.

   Example

   ```
   $ initdb -D /database/inst1 --waldir=/transaction/inst1 --lc-collate="C" --lc-ctype="C" --
   encoding=UTF8
   ```

Point

In some features, instance names are requested, and those names are required to uniquely identify the instance within the system. These features allow names that conform to WebAdmin naming conventions, so refer to the following points when determining the names:

- Maximum of 16 characters
- The first character must be ASCII alphabetic character
- The other characters must be ASCII alphanumeric characters

Note

- To balance I/O load, consider deploying the transaction log storage destination to a disk device other than the database cluster storage destination and the backup data storage destination.
- Specify "C" or "POSIX" for collation and character category. Performance deteriorates if you specify a value other than "C" or "POSIX", although the behavior will follow the rules for particular languages, countries and regions. Furthermore, this may need

to be revised when running applications on systems with different locales.
For example, specify as follows:

```
initdb --locale="C" --lc-messages="C"
initdb --lc-collate="C" --lc-ctype="C"
```

- Specify the same string in the LANG environment variable of the terminal that starts Fujitsu Enterprise Postgres as was specified in lc-messages of initdb (lc_messages of postgresql.conf). If the same string is not specified, messages displayed on the terminal that was started, as well as messages output to the log file specified in the -l option of the pg_ctl command or the postgres command used for startup, may not be output correctly.
- Specify an encoding system other than SQL_ASCII for the database. If SQL_ASCII is used, there is no guarantee that the encryption system for data in the database will be consistent, depending on the application used to insert the data.

See

Refer to "Locale Support" in "Localization" in "Server Administration" in the PostgreSQL Documentation for information on locales.

4. Set port number.

   Specify a port number in the port parameter of postgresql.conf. Ensure that the specified port number is not already used for other software. If a port number is not specified, "27500" is selected.

   Register the specified port numbers in the /etc/services file if WebAdmin is used to create other instances. WebAdmin uses the /etc/services file to check if port numbers specified as available candidates have been duplicated.

   Register any name as the service name.

5. Set the corefile output destination.

   Specify the output destination of the corefile, which can later be used to collect information for investigation, by setting the core_directory and core_contents parameters of postgresql.conf.

See

Refer to "Parameters" in the Operation Guide for information on the settings for these parameters.

6. Set the backup storage destination.

   Specify the backup data storage destination and other backup settings when backup is to be performed as a provision against database errors.

See

Refer to "Backup Methods" in the Operation Guide for information on specifying backup settings.

7. Start an instance.

   Start with the start mode of the pg_ctl command.

   If either of the following conditions are met, the message "FATAL:the database system is starting up(11189)" may be output.

   - An application, command, or process connects to the database while the instance is starting
   - An instance was started without the -W option specified

   This message is output by the pg_ctl command to check if the instance has started successfully.
   Therefore, ignore this message if there are no other applications, commands, or processes that connect to the database.

   Example

```
$ pg_ctl start -D /database/inst1
```

See

Refer to "pg_ctl" in "Reference" in the PostgreSQL Documentation for information on the pg_ctl command.

Note

If the -W option is specified, the command will return without waiting for the instance to start. Therefore, it may be unclear as to whether instance startup was successful or failed.

4.4 Configuring Remote Connections

This section describes the settings required when connecting remotely to Fujitsu Enterprise Postgres from a database application or a client command.

4.4.1 When an Instance was Created with WebAdmin

Settings related to connection

The default is to accept connections from remote computers to the database.

Change "listen_addresses" in postgresql.conf to modify the default behavior.

Refer to "Appendix D Configuring Parameters" for information on postgresql.conf.

Client Authentication Information settings

The following content is set by default when WebAdmin is used to create an instance.

- Authentication of remote connections from local machines is performed.

When changing Client Authentication Information, select [Client Authentication] from [Setting], and then change the settings.

4.4.2 When an Instance was Created with the initdb Command

Connection settings

The default setting only permits local connections from the client to the database. Remote connections are not accepted.

Change "listen_addresses" in postgresql.conf to perform remote connection.

All remote connections will be allowed when changed as shown below.

Example

```
listen_addresses = '*'
```

Also, configure the parameters shown below in accordance with the applications and number of client command connections.

Parameter name	Parameter description
superuser_reserved_connections	Number of connections reserved for database maintenance, for example backup or index rebuilding. If you need to simultaneously perform a large number of processes that exceed the default value, change this value accordingly.
max_connections	Set the value as: *numberOfSimultaneousConnectionsToInstance* + superuser_reserved_connections

Client authentication information settings

When trying to connect from a client to a database, settings are required to determine whether the instance permits connections from the client - if it does, then it is possible to make settings to determine if authentication is required.

See

Refer to "The pg_hba.conf File" in "Server Administration" in the PostgreSQL Documentation for details.

4.5 Other Settings

This section describes settings that are useful for operations.

4.5.1 Error Log Settings

This section explains the settings necessary to monitor errors in applications and operations, and to make discovering the causes easier.

Make error log settings only when instances are created with the initdb command.

When creating instances with WebAdmin, these settings are already made and hence do not need to be set.

Furthermore, some parameters are used by WebAdmin, and if changed, may cause WebAdmin to no longer work properly. Refer to "Appendix D Configuring Parameters" for details.

Note

Set the output destination for the system log to the server log so that it cannot be viewed by administrators of other instances.

Application errors are output to the system log or server log. The output destination directory for the system log and server log should have access permissions set so that they cannot be viewed by people other than the instance administrator.

Edit the following parameters in postgresql.conf:

Parameter name	Parameter description	How to enable the settings
syslog_ident	Used to specify labels to attach to messages, so that these can be identified when output to the system log if more than one Fujitsu Enterprise Postgres is used.	reload option of the pg_ctl mode
logging_collector	Specify "on" to ensure that messages are output by Fujitsu Enterprise Postgres to the server log file. The server log file is created in the log directory in the database cluster.	restart option of the pg_ctl mode
log_destination	Specify "stderr,syslog" to output messages from Fujitsu Enterprise Postgres to the screen and either the system log or the event log.	reload option of the pg_ctl mode
log_line_prefix	Specify information to be added at the start of messages output by an instance. This information is useful for automatic monitoring of messages. You can output the SQLSTATE value, output time, executing host, application name, and user ID. Refer to "What To Log" in the PostgreSQL Documentation for details. Example: log_line_prefix = '%e: %t [%p]: [%l-1] user = %u,db = %d,remote = %r app = %a '	reload option of the pg_ctl mode

Point

- If you want fewer application errors being output to the system log, refer to "When To Log" and "What To Log" in the PostgreSQL Documentation for information on how to reduce the output messages.
- If you want to separate errors output from other software, refer to "Where To Log" in the PostgreSQL Documentation to change the output destination to the server log file rather than the system log.

4.5.2 Configuring Automatic Start and Stop of an Instance

You can automatically start or stop an instance when the operating system on the database server is started or stopped.

Use the following procedure to configure automatic start and stop of an instance.

Note that, if an instance is started in a failover operation, the cluster system will control the start or stop, therefore this feature should not be used. Also, when performing database multiplexing, refer to "Enabling Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" in the Cluster Operation Guide (Database Multiplexing).

Note that "<*x*>" in paths indicates the product version.

Note

You should wait for time correction, network setup, and so on.

1. Create a unit file

   Copy the unit file sample stored in the directory below, and revise it to match the target instance.

   ```
   fujitsuEnterprisePostgresInstallDir/share/fsepsvoi.service.sample
   ```

   Example

   In the following example, the installation directory is "/opt/fsepv<*x*>server64", and the instance name is "inst1".

   ```
   # cp /opt/fsepv<x>server64/share/fsepsvoi.service.sample /usr/lib/systemd/system/
   fsepsvoi_inst1.service
   ```

   Revise the underlined portions of the options below in the unit file.

Section	Option	Specified value	Description
Unit	Description	Fujitsu Enterprise Postgres ***instanceName***	Specifies the feature overview. Specifies the name of the target instance. (*1)
Service	ExecStart	/bin/bash -c '***installDir***/bin/pgx_symstd start ***installDir dataStorageDestinationDir*** '	Command to be executed when the service is started.
	ExecStop	/bin/bash -c '***installDir***/bin/pgx_symstd stop ***installDir dataStorageDestinationDir*** '	Command to be executed when the service is stopped.
	ExecReload	/bin/bash -c '***installDir***/bin/pgx_symstd reload ***installDir dataStorageDestinationDir*** '	Command to be executed when the service is reloaded
	User	**User**	OS user account of the instance administrator.
	Group	**Group**	Group to which the instance administrator user belongs.

*1: The instance name should be as follows:

If WebAdmin is used to create the instance: *instanceName*

If the initdb command is used to create the instance: *nameThatIdentifiesTheInstance*
The naming conventions for the instance name or for identifying the instance are as follows:

- Up to 16 bytes
- The first character must be an ASCII alphabetic character
- The other characters must be ASCII alphanumeric characters

2. Enable automatic start and stop

   As the OS superuser, use the systemctl command to enable automatic start and stop.

   Example

```
# systemctl enable fsepsvoi_inst1.service
```

4.5.3 Settings when Using the Features Compatible with Oracle Databases

To use the features compatible with Oracle databases, create a new instance and execute the following command for the "postgres" and "template1" databases:

```
CREATE EXTENSION oracle_compatible;
```

Features compatible with Oracle databases are defined as user-defined functions in the "public" schema created by default when database clusters are created, so they can be available for all users without the need for special settings.

For this reason, ensure that "public" (without the double quotation marks) is included in the list of schema search paths specified in the search_path parameter.

There are also considerations for use the features compatible with Oracle databases. Refer to " Precautions when Using the Features Compatible with Oracle Databases" in the Application Development Guide for details.

4.5.4 LDAP Authentication File Settings

The LDAP authentication file refers to the following OS standard default file

/opt/openldap/ldap.conf

If you wish to use a different file, specify the LDAP authentication file you wish to set in an environment variable such as LDAPCONF, and then restart the Postgres instance. Please refer to the OpenLDAP documentation for detailed configuration details.

4.5.5 Setting the server keytab file for GSSAPI authentication

When setting the server keytab file for GSSAPI authentication, be sure to set the "krb_server_keyfile" parameter in postgresql.cocnf.

Note

The description of the "krb_server_keyfile" parameter in the "PostgreSQL Documentation" states that the default value is "FILE The default value is "FILE:/usr/local/pgsql/etc/krb5.keytab" in the "PostgreSQL Documentation", but the default value is invalid for Fujitsu Enterprise Postgres.

4.6 Integration with Message-Monitoring Software

To monitor messages output by Fujitsu Enterprise Postgres using software, configure the product to monitor SQLSTATE, instead of the message text - this is because the latter may change when Fujitsu Enterprise Postgres is upgraded.

Configure Fujitsu Enterprise Postgres to output messages in a format that can be read by the message-monitoring software by specifying "%e" in the log_line_prefix parameter of postgresql.conf to output the SQLSTATE value.

A setting example is shown below - it outputs the output time, executing host, application name, and user ID, in addition to the SQLSTATE value.

Example

```
log_line_prefix = '%e: %t [%p]: [%l-1] user = %u,db = %d,remote = %r app = %a '
```

See

Refer to "What To Log" in the PostgreSQL Documentation for information on how to configure the settings.

4.7 Setting Up and Removing OSS

This section explains how to set up OSS supported by Fujitsu Enterprise Postgres.

If you want to use OSS supported by Fujitsu Enterprise Postgres, follow the setup procedure.

If you decide not to use the OSS supported by Fujitsu Enterprise Postgres, follow the removing procedure.

To build and use OSS obtained from the web, etc., instead of OSS supported by Fujitsu Enterprise Postgres, see "4.7.10 Build with PGXS".

Information

- In this section, the applicable database that enables the features of each OSS is described as "postgres".
- Execute CREATE EXTENSION for the "template1" database also, so that each OSS can be used by default when creating a new database.

Refer to "OSS Supported by Fujitsu Enterprise Postgres" in the General Description for information on OSS other than those described below.

4.7.1 oracle_fdw

4.7.1.1 Setting Up oracle_fdw

1. Add the path of the OCI library to the environment variable. The available version of the OCI library is 11.2 or later.
   Add the installation path of the OCI library to the LD_LIBRARY_PATH environment variable.
2. As superuser, run the following command:

   ```
   $ su -
   Password:******
   # cp -r /opt/fsepv<x>server64/OSS/oracle_fdw/* /opt/fsepv<x>server64
   ```

3. If a file named libclntsh.so.11.1 does not exist in your OCI library, create a symbolic link with the name libclntsh.so.11.1 to libclntsh.so.xx.1 (xx is the version of the OCI library).

   ```
   # ln -s libclntsh.so.12.1 libclntsh.so.11.1
   ```

4. Restart Fujitsu Enterprise Postgres.
5. Execute CREATE EXTENSION for the database that will use this feature.
   Use the psql command to connect to the "postgres" database.

   ```
   postgres=# CREATE EXTENSION oracle_fdw;
   CREATE EXTENSION
   ```

Information

- If the OCI library is not installed on the server, install it using the Oracle client or Oracle Instant Client.
  Refer to the relevant Oracle manual for information on the installation procedure.

- If the version of the OCI library is updated, change the path of the OCI library in the LD_LIBRARY_PATH environment variable to the updated path. Also, re-create the symbolic link named libclntsh.so.11.1 if necessary.

Note

This feature cannot be used on instances created in WebAdmin. It can only be used via server commands.

4.7.1.2 Removing oracle_fdw

1. Execute DROP EXTENSION for the database that will use this feature.
   Use the psql command to connect to the "postgres" database.

```
postgres=# DROP EXTENSION oracle_fdw CASCADE;
DROP EXTENSION
```

2. As superuser, run the following command:

```
$ su -
Password:******
# rm -rf /opt/fsepv<x>server64/filesCopiedDuringSetup
```

Information

The files copied during setup can be checked below.

```
# find /opt/fsepv<x>server64/OSS/oracle_fdw
```

4.7.2 pg_bigm

4.7.2.1 Setting Up pg_bigm

1. Set the postgresql.conf file parameters.
   Add "pg_bigm" to the shared_preload_libraries parameter.
2. As superuser, run the following command:

```
$ su -
Password:******
# cp -r /opt/fsepv<x>server64/OSS/pg_bigm/* /opt/fsepv<x>server64
```

3. Restart Fujitsu Enterprise Postgres.
4. Execute CREATE EXTENSION for the database that will use this feature.
   Use the psql command to connect to the "postgres" database.

```
postgres=# CREATE EXTENSION pg_bigm;
CREATE EXTENSION
```

4.7.2.2 Removing pg_bigm

1. Execute DROP EXTENSION for the database that will use this feature.
   Use the psql command to connect to the "postgres" database.

```
postgres=# DROP EXTENSION pg_bigm CASCADE;
DROP EXTENSION
```

2. As superuser, run the following command:

```
$ su -
Password:******
# rm -rf /opt/fsepv<x>server64/filesCopiedDuringSetup
```

Information

The files copied during setup can be checked below.

```
# find /opt/fsepv<x>server64/OSS/pg_bigm
```

3. Set the postgresql.conf file parameters.
   Delete "pg_bigm" to the shared_preload_libraries parameter.
4. Restart Fujitsu Enterprise Postgres.

4.7.3 pg_hint_plan

4.7.3.1 Setting Up pg_hint_plan

1. Set the postgresql.conf file parameters.
   Add "pg_hint_plan" to the "shared_preload_libraries" parameter.
2. As superuser, run the following command:

```
$ su -
Password:******
# cp -r /opt/fsepv<x>server64/OSS/pg_hint_plan/* /opt/fsepv<x>server64
```

3. Restart Fujitsu Enterprise Postgres.
4. Run CREATE EXTENSION for the database that uses this feature.
   The target database is described as "postgres" here.
   Use the psql command to connect to the "postgres" database.

```
postgres=# CREATE EXTENSION pg_hint_plan;
CREATE EXTENSION
```

See

Refer to " Optimizer Hints" in the Application Development Guide for details.

4.7.3.2 Removing pg_hint_plan

Note

Unsetting pg_hint_plan will cause hints registered in the hint_plan.hints table to be lost. Therefore, it is recommended that pg_dump back up the hint_plan.hints table for each database if it is likely that pg_hint_plan will be used again later.

1. Execute DROP EXTENSION for the database that will use this feature.
   Use the psql command to connect to the "postgres" database.

```
postgres=# DROP EXTENSION pg_hint_plan CASCADE;
DROP EXTENSION
```

2. As superuser, run the following command:

```
$ su -
Password:******
# rm -rf /opt/fsepv<x>server64/filesCopiedDuringSetup
```

Information

The files copied during setup can be checked below.

```
# find /opt/fsepv<x>server64/OSS/pg_hint_plan
```

3. Set the postgresql.conf file parameters.
   Delete "pg_hint_plan" to the shared_preload_libraries parameter.
4. Restart Fujitsu Enterprise Postgres.

4.7.4 pg_dbms_stats

4.7.4.1 Setting Up pg_dbms_stats

1. Set the postgresql.conf file parameter.
   Add "pg_dbms_stats" to the "shared_preload_libraries" parameter.
2. As superuser, run the following command:

```
$ su -
Password:******
# cp -r /opt/fsepv<x>server64/OSS/pg_dbms_stats/* /opt/fsepv<x>server64
```

3. Restart Fujitsu Enterprise Postgres.
4. Run CREATE EXTENSION for the database that will use this feature.
   The target database is described as "postgres" here.
   Use the psql command to connect to the "postgres" database.

```
postgres=# CREATE EXTENSION pg_dbms_stats;
CREATE EXTENSION
```

See

Refer to "Optimizer Hints" in the Application Development Guide for details.

4.7.4.2 Removing pg_dbms_stats

Note

Unsetting pg_dbms_stats causes statistics managed by pg_dbms_stats to be lost. Therefore, it is recommended that you back up each table in the dbms_stats folder of each database in binary format if you may want to use pg_dbms_stats again later.

```
postgres > # COPY <dbms_stats Schema's table name> TO '<Filename>' FORMAT binary;
```

1. Execute DROP EXTENSION for the database that will use this feature.
   Use the psql command to connect to the "postgres" database.

```
postgres=# DROP EXTENSION pg_dbms_stats CASCADE;
DROP EXTENSION
```

2. As superuser, run the following command:

```
$ su -
Password:******
# rm -rf /opt/fsepv<x>server64/filesCopiedDuringSetup
```

Information

The files copied during setup can be checked below.

```
# find /opt/fsepv<x>server64/OSS/pg_dbms_stats
```

3. Set the postgresql.conf file parameters.
   Delete "pg_dbms_stats" to the shared_preload_libraries parameter.
4. Restart Fujitsu Enterprise Postgres.

4.7.5 pg_repack

4.7.5.1 Setting Up pg_repack

1. As superuser, run the following command:

```
$ su -
Password:******
# cp -r /opt/fsepv<x>server64/OSS/pg_repack/* /opt/fsepv<x>server64
```

2. Execute CREATE EXTENSION for the database that will use this feature.
   Use the psql command to connect to the "postgres" database.

```
postgres=# CREATE EXTENSION pg_repack;
CREATE EXTENSION
```

4.7.5.2 Removing pg_repack

1. Execute DROP EXTENSION for the database that will use this feature.
   Use the psql command to connect to the "postgres" database.

```
postgres=# DROP EXTENSION pg_repack CASCADE;
DROP EXTENSION
```

2. As superuser, run the following command:

```
$ su -
Password:******
# rm -rf /opt/fsepv<x>server64/filesCopiedDuringSetup
```

Information

The files copied during setup can be checked below.

```
# find /opt/fsepv<x>server64/OSS/pg_repack
```

4.7.6 pg_rman

4.7.6.1 Setting Up pg_rman

1. As superuser, run the following command:

```
$ su -
Password:******
# cp -r /opt/fsepv<x>server64/OSS/pg_rman/* /opt/fsepv<x>server64
```

2. Restart Fujitsu Enterprise Postgres.

Information

Before initialization of the backup catalog, it is recommended to set the parameters below in postgresql.conf. Refer to the pg_rman manual (http://ossc-db.github.io/pg_rman/index-ja.html) for details.

- log_directory
- archive_mode
- archive_command

Note

This feature cannot be used on instances created in WebAdmin. It can only be used via server commands.

4.7.6.2 Removing pg_rman

1. As superuser, run the following command:

```
$ su -
Password:******
# rm -rf /opt/fsepv<x>server64/filesCopiedDuringSetup
```

Information

The files copied during setup can be checked below.

```
# find /opt/fsepv<x>server64/OSS/pg_rman
```

2. Restart Fujitsu Enterprise Postgres.

4.7.7 pg_statsinfo

4.7.7.1 Setting Up pg_statsinfo

1. Set the postgresql.conf file parameters.
   - Add "pg_statsinfo" to the shared_preload_libraries parameter.
   - Specify the log file name for the log_filename parameter.
2. As superuser, run the following command:

```
$ su -
Password:******
# cp -r /opt/fsepv<x>server64/OSS/pg_statsinfo/* /opt/fsepv<x>server64
```

3. Restart Fujitsu Enterprise Postgres.

Information

Note that pg_statsinfo forcibly overwrites the settings below.

- log_destination

  "csvlog" is added and "stderr" is deleted.

- logging_collector

  "on" is set.

Note

This feature cannot be used on instances created in WebAdmin. It can only be used via server commands.

4.7.7.2 Removing pg_statsinfo

1. As superuser, run the following command:

```
$ su -
Password:******
# rm -rf /opt/fsepv<x>server64/filesCopiedDuringSetup
```

   ### Information

   The files copied during setup can be checked below.

```
# find /opt/fsepv<x>server64/OSS/pg_statsinfo
```

2. Set the postgresql.conf file parameters.

   - Delete "pg_statsinfo" to the shared_preload_libraries parameter.
   - Delete the log file name for the log_filename parameter.

3. Restart Fujitsu Enterprise Postgres.

4.7.8 pgBadger

4.7.8.1 Setting Up pgBadger

1. Set the postgresql.conf file parameters.
   Set the parameters so that the information required for analysis is output to the server log.
   Refer to "Documentation" in the pgBadger website (https://pgbadger.darold.net/) for details.
   The pgBadger material is stored under /opt/fsepv<x>server64/OSS/pgbadger.

2. Restart Fujitsu Enterprise Postgres.

4.7.8.2 Removing pgBadger

1. Set the postgresql.conf file parameters.
   Restores information you specified during Setup.

2. Restart Fujitsu Enterprise Postgres.

4.7.9 Pgpool-II

4.7.9.1 Setting Up Pgpool-II

1. As superuser, run the following command:

```
$ su -
Password:******
# cp -r /opt/fsepv<x>server64/OSS/Pgpool-II/* /opt/fsepv<x>server64
```

2. Execute CREATE EXTENSION for the database that will use this feature.
   Use the psql command to connect to the "postgres" database.

```
postgres=# CREATE EXTENSION pgpool_recovery;
CREATE EXTENSION
```

3. Set the postgresql.conf file parameters.
   Specify the path to pg_ctl for the pgpool.pg_ctl parameter.

4. Restart Fujitsu Enterprise Postgres.

Note

The online recovery feature of Pgpool-II cannot be used on instances created in WebAdmin. It can only be used via server commands.

4.7.9.2 Removing Pgpool-II

1. Execute DROP EXTENSION for the database that will use this feature.
   Use the psql command to connect to the "postgres" database.

```
postgres=# DROP EXTENSION pgpool_recovery CASCADE;
DROP EXTENSION
```

2. As superuser, run the following command:

```
$ su -
Password:******
# rm -rf /opt/fsepv<x>server64/filesCopiedDuringSetup
```

Information

The files copied during setup can be checked below.

```
# find /opt/fsepv<x>server64/OSS/Pgpool-II
```

3. Set the postgresql.conf file parameters.
   Delete the pg_ctl path for the pgpool.pg_ctl parameter.

4. Restart Fujitsu Enterprise Postgres.

4.7.10 Build with PGXS

Many PostgreSQL extensions are built using a build base for extensions called PGXS. Building with PGXS also generates files related to llvm. Depending on which version of llvm you are using, follow these steps:

PGXS builds also set DT_RUNPATH to the built binaries. See "4.7.10.4 Setting DT_RUNPATH" for more information.

4.7.10.1 Using the Default Version of llvm

The default version of llvm is described in "2.1 Required Operating System". If you want to use the default version of llvm, use the OSS documentation to build and install OSS.

4.7.10.2 Using a Non-Default Version of llvm

1. As superuser, copy the Makefile.global corresponding to the version of llvm you want to use. The following is an example of using version 9 of llvm. Makefile.global is overwritten when an emergency fix is applied or removed from Fujitsu Enterpise Postgres, this procedure should be performed each time a build is performed.

```
$ su -
Password:******
# cp /opt/fsepv<x>server64/lib/pgxs/src/Makefile.global-vsn9 /opt/fsepv<x>server64/lib/pgxs/src/
Makefile.global
```

2. Follow the OSS documentation to build and install OSS.

3. As superuser, run the following command:. The following is an example of using version 9 of llvm:.

```
$ su -
Password:******
# mv /opt/fsepv<x>server64/lib/bitcode/<OSS名>* /opt/fsepv<x>server64/lib/bitcode-vsn9/
```

4.7.10.3 Without llvm

If you do not use llvm, use the with _ llvm = no option when performing the build, as shown below. For other options, follow the OSS documentation.

```
# make USE_PGXS=1 with_llvm=no
```

4.7.10.4 Setting DT_RUNPATH

The default values for DT_RUNPATH are <Fujitsu Enterprise Postgres installation directory in the build environment>/lib, and $ORIGIN /../ lib.

If your build and production environments have the same Fujitsu Enterprise Postgres installation directory, you can run the built program without setting the environment variable LD_LIBRARY_PATH to <*Fujitsu Enterprise Postgres installationdirectory in the operating environment*>/lib.

If the installation directories of Fujitsu Enterprise Postgres for the build and production environments cannot be in the same location, or the production installation directory cannot be pre-determined, you can run a program built without <*Fujitsu Enterprise Postgres installationdirectory in the operating environment*>/lib in the LD_LIBRARY_PATH by doing the following:

Set the DT_RUNPATH attribute to any path.

In your production environment, create a symbolic link to <*Fujitsu Enterprise Postgres installationdirectory in the operating environment*>/lib in the appropriate path.

To do this, set the DT_RUNPATH attribute in the environment variable PG_LDFLAGS(*1).

If this is not possible, set LD_LIBRARY_PATH to <*Fujitsu Enterprise Postgres installationdirectory in the operating environment*>/lib when you run the program.

For notes on setting the environment variable LD_LIBRARY_PATH, see "When DT_RUNPATH cannot be set" in "How to Build and Run an Application that Uses Shared Libraries" in the Application Development Guide.

*1:For more information about the PG_LDFLAGS environment variable, see "Extension Building Infrastructure" in the PostgreSQL Documentation. For example, "make USE_PGXS = 1 PG_LDFLAGS =" -Wl, -rpath, '$$ORIGIN /../ libdummy', --enable-new-dtags ".

4.7.11 Build without PGXS

For extensions that do not utilize PGXS, but utilize the interface of FUJITSU Enterprise Postgres, build to explicitly set DT_RUNPATH, or set LD_LIBRARY_PATH to <FUJITSU Enterprise Postgres installation directory>/lib at runtime.

For information about how to set DT_RUNPATH, refer to "Setting DT_RUNPATH for Applications" in the Application Development Guide.

For notes on using LD_LIBRARY_PATH without setting DT_RUNPATH, refer to "When DT_RUNPATH cannot be set" in "How to Build and Run an Application that Uses Shared Libraries" in the Application Development Guide.

4.8 Deleting Instances

This section explains how to delete an instance.

- 4.8.1 Using WebAdmin
- 4.8.2 Using Server Commands

Note

- Always use WebAdmin to delete instances that were created or imported using WebAdmin. Because WebAdmin management information cannot be deleted, WebAdmin will determine that the instance is abnormal.
- If you have set automatic start and stop of the instance, execute the following commands to disable the script and cancel registration.

  ```
  systemctl disable nameOfUnitFileThatPerformsAutomaticStartAndStop
  rm /usr/lib/systemd/system/nameOfUnitFileThatPerformsAutomaticStartAndStop
  ```

  Example

  ```
  # systemctl disable fsepsvoi_inst1.service
  # rm /usr/lib/systemd/system/fsepsvoi_inst1.service
  ```

4.8.1 Using WebAdmin

This section explains how to delete an instance using WebAdmin.

Use the following procedure to delete an instance.

1. Stop the instance

   In the [Instances] tab, select the instance to stop and click .

2. Back up files.

   Before deleting the instance, back up any required files under the data storage destination, the backup data storage destination, and the transaction log storage destination.

3. Delete the instance

   In the [Instances] tab, select the instance to delete and click .

Note

Deleting an instance deletes only the following lowest-level directories. If they are not required, delete them manually.

- Data storage destination
- Backup data storage destination
- Transaction log storage destination (if different from the data storage destination)

4.8.2 Using Server Commands

This section explains how to delete an instance using server commands.

Use the following procedure to delete an instance.

1. Stop the instance

   Execute the stop mode of the pg_ctl command.

   An example is shown below:

   Example

   ```
   $ pg_ctl stop -D /data/inst1
   ```

2. Back up files.

   Before deleting the instance, back up any required files under the data storage destination, the backup data storage destination, and the transaction log storage destination.

3. Delete the instance

   Use a standard UNIX tool (the rm command) to delete the following directories:

   - Data storage destination
   - Backup data storage destination
   - Transaction log storage destination (if a directory different from the data storage directory was specified)

Chapter 5 Uninstallation

This chapter describes the procedure for uninstalling Fujitsu Enterprise Postgres.

5.1 Uninstallation in Interactive Mode

Uninstall according to the following procedure:

Note that "*x*SP*z*" in sample windows indicates the version and level of products to uninstall and "<*x*>" in paths indicates the product version.

Note

- All files and directories in the installation directory are deleted during uninstallation. If user files have been placed in the installation directory, back them up before uninstallation if necessary.
- On SLES, before uninstallation, ensure that JRE 8 is installed, and export the JAVA_HOME environment variable.

```
#export JAVA_HOME="Jre8InstallDir"
```

- To reinstall Fujitsu Enterprise Postgres after it was uninstalled, and reuse an instance that was already created so that it can be managed from WebAdmin, back up the directory shown below in which the WebAdmin instance management information had been defined before uninstalling Fujitsu Enterprise Postgres, and then restore the backed up directory to its original location once Fujitsu Enterprise Postgres has been reinstalled.

  Follow the procedure below to perform the backup.

  1. Stop the WebAdmin server. Refer to "B.1.3 Stopping the Web Server Feature of WebAdmin" for details.
  2. Back up the following directory:

     webAdminInstallDir/data/fepwa

Information

If an error occurs while the product is being uninstalled, refer to "Uninstall (middleware) Messages" in the Fujitsu Enterprise Postgres product website, and take the required action.

1. Delete the operation information

If the Fujitsu Enterprise Postgres operation information has been registered in the operating system or another middleware product, for example, then it must be deleted. Cases in which deletion is required are as follows:

- For a cluster system using failover operation integrated with PRIMECLUSTER, stop the cluster applications and delete the cluster application resources and Fujitsu Enterprise Postgres resources. Refer to the Cluster Operation Guide (PRIMECLUSTER) and PRIMECLUSTER Installation and Administration Guide for details.
- If you have set automatic start and stop of the instance, execute the following commands to disable the script and cancel registration.

  ```
  systemctl disable nameOfUnitFileThatPerformsAutomaticStartAndStop
  rm /usr/lib/systemd/system/nameOfUnitFileThatPerformsAutomaticStartAndStop
  ```

  Example

  ```
  # systemctl disable fsepsvoi_inst1.service
  # rm /usr/lib/systemd/system/fsepsvoi_inst1.service
  ```

2. Stop applications and programs

Before starting the uninstallation, stop the following:

- Applications that use the product
- Connection Manager
- Instance

  Using WebAdmin

  In the [Instances] tab, select the instance to stop and click .

  Using server commands

  Execute the pg_ctl command in stop mode.

  ```
  $ /opt/fsepv<x>server64/bin/pg_ctl stop -D /database/inst1
  ```

- Web server feature of WebAdmin

  Execute the WebAdminStop command to stop the Web server feature of WebAdmin.

  Example

  If WebAdmin is installed in /opt/fsepv<*x*>webadmin:

  ```
  # cd /opt/fsepv<x>webadmin/sbin
  # ./WebAdminStop
  ```

- Mirroring Controller

  Execute the mc_ctl command with the stop mode option specified and stop the Mirroring Controller.

  Example

  ```
  $ mc_ctl stop -M /mcdir/inst1
  ```

- pgBadger
- Pgpool-II

3. Start the Uninstall (middleware)

Execute the following command:

```
# /opt/FJSVcir/cimanager.sh -c
```

4. Select the software

Type the number for the product to be uninstalled, and press Enter.

5. Start the uninstallation

To start the uninstallation, type "y" and press Enter.

To display the list of products again, type "b" and press Enter.

6. Finish the uninstallation

Upon successful completion, the window below is displayed.

The installation directory may remain after uninstallation. If it is not required, delete it.

```
Uninstalling...

productName is being uninstalled now.
    100% ##############################


The following products have been uninstalled successfully:
    productName
```

```
Uninstallation of "productName" has completed successfully.



Exiting Uninstaller.
```

5.2 Uninstallation in Silent Mode

Uninstall according to the following procedure:

Note

- All files and directories in the installation directory are deleted during uninstallation. If user files have been placed in the installation directory, back them up before uninstallation if necessary.
- On SLES, before uninstallation, ensure that JRE 8 is installed, and export the JAVA_HOME environment variable.

```
#export JAVA_HOME="Jre8InstallDir"
```

- To reinstall Fujitsu Enterprise Postgres after it was uninstalled, and reuse an instance that was already created so that it can be managed from WebAdmin, back up the directory shown below in which the WebAdmin instance management information had been defined before uninstalling Fujitsu Enterprise Postgres, and then restore the backed up directory to its original location once Fujitsu Enterprise Postgres has been reinstalled.

  Follow the procedure below to perform the backup.

  1. Stop the WebAdmin server. Refer to "B.1.3 Stopping the Web Server Feature of WebAdmin" for details.
  2. Back up the following directory:

     webAdminInstallDir/data/fepwa

See

Refer to the Fujitsu Enterprise Postgres product website for information on uninstallation in silent mode, such as the error messages.

1. Delete the operation information

If the Fujitsu Enterprise Postgres operation information has been registered in the operating system or another middleware product, for example, then it must be deleted. Cases in which deletion is required are as follows:

- For a cluster system using failover operation integrated with PRIMECLUSTER, stop the cluster applications and delete the cluster applications resources and Fujitsu Enterprise Postgres resources. Refer to the Cluster Operation Guide (PRIMECLUSTER) and PRIMECLUSTER Installation and Administration Guide for details.
- If you have set automatic start and stop of the instance, execute the following commands to disable the script and cancel registration.

  ```
  systemctl disable nameOfUnitFileThatPerformsAutomaticStartAndStop
  rm /usr/lib/systemd/system/nameOfUnitFileThatPerformsAutomaticStartAndStop
  ```

  Example

  ```
  # systemctl disable fsepsvoi_inst1.service
  # rm /usr/lib/systemd/system/fsepsvoi_inst1.service
  ```

2. Stop applications and programs

Before starting the uninstallation, stop the following:

- Applications that use the product
- Connection Manager
- Instance

  Using WebAdmin

  In the [Instances] tab, select the instance to stop and click .

  Using server commands

  Execute the pg_ctl command with the stop mode option specified.

```
$ /opt/fsepv<x>server64/bin/pg_ctl stop -D /database/inst1
```

- Web server feature of WebAdmin

  Execute the WebAdminStop command to stop the Web server feature of WebAdmin.

  Example

  If WebAdmin is installed in /opt/fsepv<*x*>webadmin:

```
# cd /opt/fsepv<x>webadmin/sbin
# ./WebAdminStop
```

- Mirroring Controller

  Execute the mc_ctl command with the stop mode option specified and stop the Mirroring Controller.

  Example

```
$ mc_ctl stop -M /mcdir/inst1
```

- pgBadger
- Pgpool-II

3. Run the uninstallation

Execute the command below.

The installation directory may remain after uninstallation. If it is not required, delete it.

Example

```
# /opt/fsepv<x>server64/setup/suninst.sh
```

In the example above, /opt/fsepv<*x*>server64/setup is the name of the installation directory in which the server product is installed.

```
# /opt/fsepv<x>webadmin/setup/suninst.sh
```

In the example above, /opt/fsepv<*x*>webadmin/setup is the name of the installation directory in which the WebAdmin is installed.

```
# /opt/fsepv<x>pgpool-II/setup/suninst.sh
```

In the example above, /opt/fsepv<*x*>pgpool-II/setup is the name of the installation directory in which the Pgpool-II is installed.

Appendix A Recommended WebAdmin Environments

This appendix describes the recommended WebAdmin environment. The following explanation is based on the assumption that Microsoft Edge is used unless otherwise stated.

Note

The displayed screen varies depending on your environment, so check and set according to the screen.

A.1 Recommended Browser Settings

- Use a display resolution of 1280 x 768 or higher, and 256 colors or more.
- Select [Setting] >>[Appearance] >> [Font size] >> [Medium (Recommended)].
- Select [Setting] >>[Appearance] >> [Zoom] >> [100%].

A.2 How to Set Up the Pop-up Blocker

If the Pop-up Blocker is enabled, use the procedure below to configure settings to allow pop-ups from the server where Fujitsu Enterprise Postgres is installed.

1. Click [Setting] >> [Cookie and site permissions] >> [All Permissions] >> [Pop-ups and redirects].
   If the [Block (Recommended)] switch is not on (blue), the pop-up blocker is not working, and no further action is required.
2. Under [Pop-ups and Redirects], click the [Allow] >> [Add] button.
3. In [Add Site], in [Site], enter the address of the server where you installed Fujitsu Enterprise Postgres and click the [Add] button.
4. Close Microsoft Edge.

Appendix B Setting Up and Removing WebAdmin

This appendix describes how to set up and remove WebAdmin.

Note that "<*x*>" in paths indicates the product version.

B.1 Setting Up WebAdmin

This section explains how to set up WebAdmin.

B.1.1 Setting Up WebAdmin

Follow the procedure below to set up WebAdmin.

1. Change to the superuser

   Acquire superuser privileges on the system.

   Example

   ```
   $ su -
   Password:******
   ```

2. Set up WebAdmin

   Set up WebAdmin.

   Example

   If WebAdmin is installed in /opt/fsepv<*x*>webadmin:

   ```
   # cd /opt/fsepv<x>webadmin/sbin
   # ./WebAdminSetup
   ```

3. Specify the port number

   Specify the following port numbers to be used in WebAdmin.

   Refer to the "/etc/services" file and only change to a different port number if there is overlap with a port number from another service.

   Make a note of the port number for the Web server, because it will be required for starting the WebAdmin window.

Item	Value (recommended value)
Web server port number enter port number of Web Server (default: 27515):	27515
WebAdmin internal port number enter Internal port number for WebAdmin (default: 27516):	27516
WebAdmin automatic start Start WebAdmin automatically when system starting? [y,n] (default: y)	y

Web server port number

Specify a numeric value from 1024 to 32767 for the port number to be used for communication between the Web browser and the Web server.

The Web server port number will be registered as a port number with the following service name in the "/etc/services" file.

fsep_150_WA_64_WebAdmin_Port1

WebAdmin internal port number

Specify a numeric value from 1024 to 32767 for the port number to be used for communication between the Web server and the WebAdmin runtime environment.

The WebAdmin internal port number will be registered as a port number with the following service name in the /etc/services file.

fsep_140_WA_64_WebAdmin_Port2

WebAdmin automatic start

Select whether or not to start WebAdmin when the machine is started.

Note

- Unused port numbers
  Irrespective of the information specified in the "/etc/services" file, unused port numbers in the OS and other products can sometimes be automatically numbered and then used, or port numbers specified in environment files within products may also be used. Check the port numbers used by the OS and other products, and ensure that these are not duplicated.
- Access restrictions
  Prevent unauthorized access and maintain security by using a firewall product, or the packet filtering feature of a router device, to restrict access to the server IP address and the various specified port numbers.
- Port access permissions
  If a port is blocked (access permissions have not been granted) by a firewall, enable use of the port by granting access. Refer to the vendor document for information on how to grant port access permissions.
  Consider the security risks carefully when opening ports.
- Changing port numbers
  When using WebAdmin in multiserver mode, it is recommended not to change WebAdmin ports after creating instances. Otherwise, the created instances may not be accessible through WebAdmin after the port is changed.

B.1.2 Starting the Web Server Feature of WebAdmin

Follow the procedure below to start the Web server feature of WebAdmin.

1. Change to the superuser

   Acquire superuser privileges on the system.

   Example

   ```
   $ su -
   Password:******
   ```

2. Start the Web server feature of WebAdmin

   Execute the WebAdminStart command to start the Web server feature of WebAdmin.

   Example

   If WebAdmin is installed in /opt/fsepv<*x*>webadmin:

   ```
   # cd /opt/fsepv<x>webadmin/sbin
   # ./WebAdminStart
   ```

B.1.3 Stopping the Web Server Feature of WebAdmin

Follow the procedure below to stop the Web server feature of WebAdmin.

1. Change to the superuser

   Acquire superuser privileges on the system.

   Example

   ```
   $ su -
   Password:******
   ```

2. Stop the Web server feature of WebAdmin

   Execute the WebAdminStop command to stop the Web server feature of WebAdmin.

   Example

   If WebAdmin is installed in /opt/fsepv<*x*>webadmin:

```
# cd /opt/fsepv<x>webadmin/sbin
# ./WebAdminStop
```

Note

- For efficient operation of WebAdmin, it is recommended that the Web server feature be stopped only during a scheduled maintenance period.
- When WebAdmin is used to create and manage instances in a multiserver configuration, the Web server feature must be started and running on all servers at the same time.

B.2 Removing WebAdmin

This section explains how to remove WebAdmin.

This removal procedure stops WebAdmin and ensures that it no longer starts automatically when the machine is restarted.

1. Change to the superuser

   Acquire superuser privileges on the system.

   Example

```
$ su -
Password:******
```

2. Remove WebAdmin setup

   Execute the WebAdminSetup command to remove WebAdmin setup.

   Example

   If WebAdmin is installed in /opt/fsepv<*x*>webadmin:

```
# cd /opt/fsepv<x>webadmin/sbin
# ./WebAdminSetup -d
```

B.3 Using an External Repository for WebAdmin

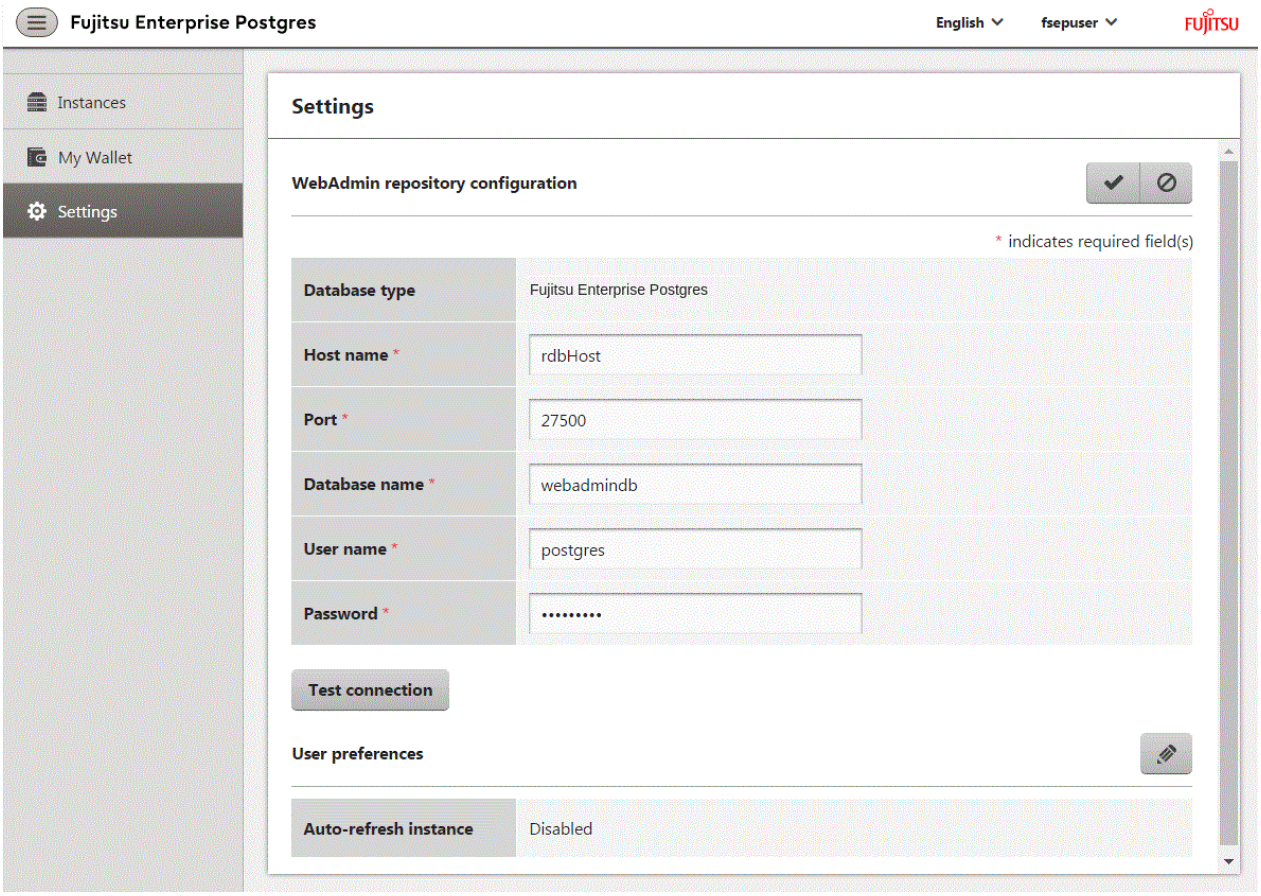
WebAdmin can be configured to use an external database, where it can store the various metadata information it uses. WebAdmin will use this database as a repository to store the information it uses to manage all the created instances. This can be a Fujitsu Enterprise Postgres database or an Open Source PostgreSQL V9.2 or later database.

Using an external database as a WebAdmin repository provides you with more flexibility in managing WebAdmin. This repository can be managed, backed up and restored as needed using pgAdmin or command line tools, allowing users to have greater flexibility and control.

Follow the procedure below to set up the repository.

1. Start WebAdmin, and log in to the database server.

2. Click the [Settings] tab, and then click in the [WebAdmin repository configuration] section.



3. Enter the following items:

   - [Host name]: Host name of the database server
   - [Port]: Port number of the database server
   - [Database name]: Name of the database
   - [User name]: User name to access the database
   - [Password]: Password of the database user

Note

- Database type
  It is recommended to use a Fujitsu Enterprise Postgres database as a repository. A compatible PostgreSQL database can also be used as an alternative.
- It is recommended to click [Test connection] to ensure that the details entered are valid and WebAdmin is able to connect to the target database.
- Host name, Database name, User name, Password should not contain hazardous characters. Refer to "Appendix C WebAdmin Disallow User Inputs Containing Hazardous Characters".

4. Click to register the repository details.

Note

- Once the repository is set up, it can be changed any number of times by the user logged into WebAdmin. When a repository is changed:
  - It is recommended to preload the backup into this database.
  - If the data is not preloaded, WebAdmin will create a new repository.
- The database repository can be set up even after WebAdmin was already used to create instances. In that scenario, the instances already created are retained and can continue to be operated on.
- If the instance used as a repository is stopped, WebAdmin will be unusable. For this reason, it is recommended to be familiar with starting an instance from the command line. If the instance is stopped for any reason, start it from the command line and WebAdmin will be usable again.

B.4 Using the WebAdmin Auto-Refresh Feature

The WebAdmin auto-refresh feature automatically refreshes the operating status of all instances in the Instance list at the specified interval. It also refreshes the details of the selected instance.

Follow the procedure below to configure the auto-refresh options.

1. Click the [Settings] tab, and then click in the [User preferences] section.
2. Enter the following items:
   - [Auto-refresh instance]: To use the auto-refresh feature, select "Enabled". The default is "Disabled".
   - [Refresh interval (seconds)]: Number of seconds between each refresh. This is a countdown timer, which is reset every time the instance status is refreshed by any operation. Specify a value from 30 to 3600 (seconds). The default is 30.
3. Click to save the auto-refresh settings.

Note

- Auto-refresh will run only if the [Instances] page is displayed and no user-initiated operation is in progress.
- A text indicator, which is independent of auto-refresh, is displayed at the top of the Instance list. It is dynamically updated to display when the page was last refreshed.

Appendix C WebAdmin Disallow User Inputs Containing Hazardous Characters

WebAdmin considers the following as hazardous characters, which are not allowed in user inputs.

```
| (pipe sign)

& (ampersand sign)

; (semicolon sign)

$ (dollar sign)

% (percent sign)

@ (at sign)

' (single apostrophe)

" (quotation mark)

\' (backslash-escaped apostrophe)

\" (backslash-escaped quotation mark)

<> (triangular parenthesis)

() (parenthesis)

+ (plus sign)

CR (Carriage return, ASCII 0x0d)

LF (Line feed, ASCII 0x0a)

, (comma sign)

\ (backslash)
```

Appendix D Configuring Parameters

WebAdmin operates and manages databases according to the contents of the following configuration files:

- postgresql.conf

  Contains various items of information that define the operating environment of Fujitsu Enterprise Postgres.

- pg_hba.conf

  Contains various items of information related to client authentication.

These configuration files are deployed to a data storage destination. Data is written to them when the instance is created by WebAdmin and when settings are changed, and data is read from them when the instance is started and when information from the [Setting] menu is displayed.

Direct editing of each configuration file is possible with a text editor.

See

Refer to "Server Configuration" and "Client Authentication" in "Server Administration" in the PostgreSQL Documentation for information on the parameters.

Note

WebAdmin checks for port number and backup storage path anomalies when various operations are performed. An anomaly occurs when the value of [Port number] and/or [Backup storage path] in WebAdmin is different from the value of the corresponding parameter in postgresql.conf. Refer to "Anomaly Detection and Resolution" in the Operation Guide for details.

postgresql.conf

Parameters that can be changed in WebAdmin

The postgresql.conf parameters that can be changed in WebAdmin are shown below:

Section	WebAdmin item	postgresql.conf file parameter
Instance Configuration		
Character encoding	Character set	client_encoding
	Message locale	lc_messages
Communication	Max connections	max_connections
SQL options	Transform NULL format	transform_null_equals
	Date output format	DateStyle (*1)
	Interval output format	IntervalStyle
	Number of digits for floating values	extra_float_digits
	Transaction isolation levels	default_transaction_isolation
	Currency format	lc_monetary
	Date and time format	lc_time
	Numerical value format	lc_numeric
Memory	Sort memory (KB)	work_mem
	Shared buffers (KB)	shared_buffers
Streaming replication	WAL level	wal_level

Section	WebAdmin item	postgresql.conf file parameter
	Maximum WAL senders	max_wal_senders
	WAL save size (MB)	wal_keep_size
	Hot standby	hot_standby
	Synchronous standby names	synchronous_standby_names
	WAL receiver timeout (ms)	wal_receiver_timeout
Edit instance		
	Instance name	n/a
	Instance port	port
	Backup storage path	backup_destination

*1: If you specify "Postgres" as the output format, dates will be output in the "12-17-1997" format, not the "Wed Dec 17 1997" format used in the PostgreSQL Documentation.

Information

- Calculate the maximum number of connections using the formula below:

```
maximumNumberOfConnections = maximumNumberOfConnectionsFromApplications + 3 (*1)
```

*1: 3 is the default number of connections required by the system.
Calculate the maximum number of connections using the following formula when changing superuser_reserved_connections (connections reserved for use by the superuser) in postgresql.conf.

```
maximumNumberOfConnections = maximumNumberOfConnectionsFromApplications +
superuser_reserved_connections
```

- Also check if the memory used exceeds the memory installed (refer to "Parameters automatically set by WebAdmin according to the amount of memory").
- When modifying "Shared buffers" or "Max connections", edit the kernel parameter. Refer to "Appendix H Configuring Kernel Parameters", and "Managing Kernel Resources" in "Server Administration" in the PostgreSQL Documentation for details.

Parameters set by WebAdmin

The following postgresql.conf parameters are set by WebAdmin during instance startup (they will be ignored even if specified in postgresql.conf):

Parameter	Value
listen_addresses	*
log_destination	'stderr,syslog'
logging_collector	on
log_line_prefix	'%e: %t [%p]: [%l-1] user = %u,db = %d,remote = %r app = %a '
log_filename (*1) (*2)	'logfile-%a.log'
log_file_mode	0600
log_truncate_on_rotation	on
log_rotation_age	1d

*1: The server logs are split into files based on the day of the week, and are rotated after each week.

*2: If the date changes while the instance is stopped, old logs are not deleted and continue to exist.
Manually delete old logs that are no longer required to release disk space.

Parameters automatically set by WebAdmin according to the amount of memory

The postgresql.conf parameters automatically set according to the amount of installed memory, during the creation of instances by WebAdmin, are shown below:

Parameter	Value
shared_buffers	30% of the machine's installed memory
work_mem	30% of the machine's installed memory / max_connections / 2
effective_cache_size	75% of the machine's installed memory
maintenance_work_mem	10% of the machine's installed memory / (1 + autovacuum_max_workers) (*1)

*1: The value will be capped at 2097151 KB.

When determining the values to be configured in the above parameters, you must take into account any anticipated increases in access volume or effects on performance during business operations, such as the number of applications and commands that will access the instance, and the content of processes. Also, note that in addition to Fujitsu Enterprise Postgres, other software may be running on the actual database server. You will need to determine the degree of priority for the database and other software, as well as the memory allocation size.

WebAdmin automatically configures complex parameter settings such as those mentioned above, based on the size of the internal memory of the machine. This enables maximum leverage of the machine memory to facilitate resistance against fluctuations during business operations.

Accordingly, the effects of the above-mentioned factors must be estimated and taken into account when determining and configuring parameter values, so that memory resources can be effectively allocated among other software or instances, and so that adverse effects can be mutually avoided. Refer to "Memory" in "Resource Consumption", and "Planner Cost Constants" in "Query Planning", under "Server Administration" in the PostgreSQL Documentation for information on parameter values and required considerations.

Parameter values can be modified using the WebAdmin [Setting] menu, or edited directly using a text editor.

If adding an instance, determine the parameter values, including for existing instances, and make changes accordingly.

See

Kernel parameters need to be tuned according to the parameters being changed. Refer to "Appendix H Configuring Kernel Parameters", and "Managing Kernel Resources" in "Server Administration" in the PostgreSQL Documentation for information on tuning kernel parameters.

Note

- Do not directly edit the following postgresql.conf parameters with a text editor, otherwise WebAdmin may not work properly if you make a mistake):
  - archive_mode
  - archive_command
  - wal_level
  - log_line_prefix
  - log_destination
  - logging_collector
  - log_directory
  - log_file_mode
  - log_filename

    - log_truncate_on_rotation
    - log_rotation_age
- You must take care with the following parameter:
    - superuser_reserved_connections

      Set it to a number that includes the 3 connections required in WebAdmin (the default is 3).

pg_hba.conf

Refer to "Client Authentication" in "Server Administration" in the PostgreSQL Documentation for information on content that can be configured in pg_hba.conf.

- Configure the instance administrator permissions in the "local" connection format settings. WebAdmin may not work properly if permissions are not configured.
- If you specify an item or value that cannot be set by WebAdmin when editing the pg_hba.conf file with a text editor, it will not be possible to reference that line from WebAdmin.

Appendix E Estimating Database Disk Space Requirements

This appendix describes how to estimate database disk space requirements.

E.1 Estimating Table Size Requirements

The following tables provide the formulas for estimating table size requirements.

Table E.1 Estimation formula when the record length is 2032 bytes or less

Item	Estimation formula (bytes)
(1) Record length	27(*1) + NULL map + OID + column data NULL map: Number of columns / 8 (*2) OID: 4 Column data: Sum of column lengths *1: Record header section *2: Round the result up to the next integer. - Because the column data is placed in boundaries of 8 bytes, you need to make an adjustment so that the sum of the record header section, NULL map and OID is a multiple of 8. For example, if the calculated length is 27 + 1 / 8 (rounded up) + 0 = 28 bytes, add 4 to make the length 32 bytes. - Because the data of each column is placed in boundaries of the defined data type, take the boundary of each data type into account for the length of the column data. For example, the length of the column data in the table below will not be the sum of the data types, which is 37 bytes, but will instead be 64 bytes following boundary adjustment. Definition: create table tb1(c1 char(1), c2 bigint, c3 int, c4 box) Estimation: CHAR type 1 byte + boundary adjustment of 7 bytes for BIGINT type 8 bytes + BIGINT type 8 bytes + INT type 4 bytes + boundary adjustment of 12 bytes for BOX type 32 bytes + BOX type 32 bytes = 64 bytes - Because each record is placed in boundaries of 8 bytes, you need to make an adjustment so that the length of the column data is a multiple of 8. - If the calculated record length exceeds 2,032 bytes, the variable length data in the record might be compressed automatically. If so, use the estimation formulas in "Table E.2 Estimation formula when the record length exceeds 2032 bytes" to estimate the table size.
(2) Page size requirement	8192 (*1) × fillfactor (*2) - 24 (*3) *1: Page length (8192) *2: Value of the fillfactor specified in the table definitions (if omitted, 100%) *3: Page header (24) - The calculated (2) page size requirement will be rounded down to the nearest integer.
(3) Number of records per page	(2) Page size requirement / ((1) record length + 4 (*1)) *1: Pointer length (4) - The result will be rounded down to the nearest integer.

Item	Estimation formula (bytes)
(4) Number of pages required for storing records	Total number of records / (3) number of records per page - The result will be rounded up to the next integer.
(5) Amount of space	(4) Number of pages required for storing records x page length x safety factor (*1) *1: Specify 2.0 or higher. - This is the safety factor assumed if vacuuming is performed for garbage collection in tables and indexes.

Table E.2 Estimation formula when the record length exceeds 2032 bytes

Item	Estimation formula (bytes)
(5) Amount of space	Total number of records x (1) record length x safety factor (*1) *1: Specify 2.0 or higher. - This is the safety factor assumed if vacuuming is performed for garbage collection in tables and indexes.

E.2 Estimating Index Size Requirements

This section provides the formulas for estimating index size requirements.

Fujitsu Enterprise Postgres provides six index types: B-tree, Hash, GiST, GIN, SP-GiST, and VCI. If you do not specify the index type in the CREATE INDEX statement, a B-tree index is generated.

The following describes how to estimate a B-tree index. Refer to "E.7 Estimating VCI Disk Space Requirements" for information on how to estimate VCI.

A B-tree index is saved as a fixed-size page of 8 KB. The page types are meta, root, leaf, internal, deleted, and empty. Since leaf pages usually account for the highest proportion of space required, you need to calculate the requirements for these only.

Table E.3 Estimation formula when the key data length is 512 bytes or less

Item	Estimation formula (bytes)
(1) Entry length	8 (*1) + key data length (*2) *1: Entry header *2: The key data length depends on its data type (refer to "E.3 Sizes of Data Types" for details). Because each entry is placed in boundaries of 8 bytes, you need to make an adjustment so that the length of the key data is a multiple of 8. For example, if the calculated length is 28 bytes, add 4 to make the length 32 bytes. - If the key data length exceeds 512 bytes, key data may be automatically compressed. In this case, use the estimation formula given in "Table E.4 Estimation formula when the key data length exceeds 512 bytes" to estimate the key data length.
(2) Page size requirement	8192 (*1) × fillfactor (*2) - 24 (*3) - 16 (*4) *1: Page length (8192) *2: Value of the fillfactor specified in the index definitions (if omitted, 90%) In the case of indexes of primary key constraints and unique constraints, the value of the fillfactor specified for each constraint in the table definitions (if omitted, 90%) *3: Page header (24) *4: Special data (16)

Item	Estimation formula (bytes)
	- The calculated (2) page size requirement will be rounded down to the nearest integer.
(3) Number of entries per page	(2) Page size requirement / ((1) entry length + 4 (*1)) *1: Pointer length - Result of (3) number of entries per page will be rounded down to the nearest integer.
(4) Number of pages required for storing indexes	Total number of records / (3) number of entries per page - Result of (4) number of pages required for storing indexes will be rounded up to the nearest integer.
(5) Space requirement	(4) Number of pages required for storing indexes x 8192 (*1) / usage rate (*2) *1: Page length *2: Specify 0.7 or lower.

Table E.4 Estimation formula when the key data length exceeds 512 bytes

Item	Estimation formula (bytes)
(5) Space requirement	Total number of records x key data length x compression ratio (*1) / usage rate (*2) *1: The compression ratio depends on the data value, so specify 1. *2: Specify 0.7 or lower as the usage rate.

E.3 Sizes of Data Types

This section lists the sizes of the data types.

E.3.1 Sizes of Fixed-Length Data Types

The following table lists the sizes of fixed-length data types.

Data type	Size (bytes)
SMALLINT (INT2)	2
INTEGER (INT4)	4
BIGINT (INT8)	8
REAL	4
DOUBLE PRECISION	8
SERIAL (SERIAL4)	4
BIGSERIAL (SERIAL8)	8
MONEY	8
FLOAT	8
FLOAT (1-24)	4
FLOAT (25-53)	8
TIMESTAMP WITHOUT TIME ZONE	8
TIMESTAMP WITH TIME ZONE	8
DATE	4
TIME WITHOUT TIME ZONE	8
TIME WITH TIME ZONE	12

Data type	Size (bytes)
INTERVAL	12
BOOLEAN	1
CIDR	IPv4: 7 IPv6: 19
INET	IPv4: 7 IPv6: 19
MACADDR	6
MACADDR8	8
POINT	16
LINE	32
LSEG	32
BOX	32
CIRCLE	24

E.3.2 Sizes of Variable-Length Data Types

The following table lists the sizes of variable-length data types.

Data type	Size (bytes)	Remarks
path	Length of size portion + 12 + 16 x number of vertices	1) When carrying out division, round to the next integer. 2) If the real data length is less than 127, then the length of the size portion is 1 byte, otherwise it is 4 bytes. 3) The number of bytes per character depends on the character set (refer to "E.3.4 Number of Bytes per Character" for details).
polygon	Length of size portion + 36 + 16 x number of vertices	
decimal	Length of size portion + 2 + (integer precision / 4 + decimal precision / 4) x 2	
numeric		
bytea	Length of size portion + real data length	
character varying(*n*), varchar(*n*)	Length of size portion + number of characters x number of bytes per character	
character(*n*), char(*n*)	Length of size portion + *n* x number of bytes per character	
text	Length of size portion + number of characters x number of bytes per character	

E.3.3 Sizes of Array Data Types

The following table lists the sizes of array data types.

Data type	Size (bytes)	Remarks
Array	Length of size portion + 12 + 8 x number of dimensions + data size of each item	If the real data length is less than 127, then the length of the size portion is 1 byte, otherwise it is 4 bytes. - Example of estimation when array data is "ARRAY[[1,2,3], [1,2,3]]" Number of dimensions: 2 INTEGER data size: 4 Total size = 1+12+8x2+6x4 = 53

E.3.4 Number of Bytes per Character

The following table lists the number of bytes per character.

The given values relate to the common character sets EUC-JP and UTF8.

Character type	Character set	Number of bytes per character
ASCII	EUC_JP	1
Halfwidth katakana	EUC_JP	2
JIS X 0208 kanji characters	EUC_JP	2
JIS X 0212 kanji characters	EUC_JP	3
ASCII	UTF8	1
Halfwidth katakana	UTF8	3
JIS X 0208 kanji characters	UTF8	3
JIS X 0212 kanji characters	UTF8	3

E.4 Estimating Transaction Log Space Requirements

This section provides the formula for estimating transaction log space requirements.

```
Transaction log space requirements = max_wal_size
```

However, if the update volume is extremely high (for example, due to a large data load and batch processing), disk writing at a checkpoint may not be able to keep up with the load, and a higher number of transaction logs than indicated here may temporarily be accumulated.

E.5 Estimating Archive Log Space Requirements

This section explains how to estimate archive log space requirements.

The archive log is an archive of the transaction logs from the time of a previous backup to the present, so it fluctuates depending on the backup period and the content of update transactions.

The longer the backup period and the more update transactions, the greater the space required for the archive log.

Therefore, measure the actual archive log space by using a test environment to simulate backup scheduling and database update in a real operating environment.

E.6 Estimating Backup Disk Space Requirements

This section provides the formula for estimating backup disk space requirements.

```
Backup disk space requirements = size of the database cluster x 2 + transaction log space requirements
                               + archive log space requirements
```

If the pgx_dmpall command performs a backup using a user exit, the backup disk size differs according to the database resources targeted for backup and the copy method.

E.7 Estimating VCI Disk Space Requirements

This section provides the formula for estimating VCI disk space requirements.

```
Disk space = (number of rows in tables) x (number of bytes per row) x (compression ratio) + (WOS size)
```

Number of bytes per row

```
Number of bytes per row = (19 + (number of columns specified in CREATE INDEX) / 8
                                  + (number of bytes per single column value)) x 1.1
```

Note: Round up the result to the nearest integer.

Compression ratio

Specify a value between 0 and 1. Since compression ratio depends on the data being compressed, use actual data or test data that simulates it, then compare the value with the estimation result. As a guide, the compression ratio measured with the Fujitsu sample data is shown below:

- Data with high degree of randomness (difficult to compress): Up to approximately 0.9 times.
- Data with high degree of similarity (easy to compress): Up to approximately 0.5 times.

WOS size

```
WOS size = (number of WOS rows) / 185 x 8096
```

One row is added to the number of WOS rows for each INSERT and DELETE, and two rows are added for UPDATE. On the other hand, the number decreases to 520,000 rows or less during conversion to ROS performed by the ROS control daemon.

Note

VCI does not support retrieval of disk space usage using the database object size function pg_indexes_size. To find out the actual total VCI disk space, check the disk space of the storage directory using an OS command or other method.

Appendix F Estimating Memory Requirements

This appendix explains how to estimate the memory.

F.1 Fujitsu Enterprise Postgres Memory Requirements

This section describes the formulas for estimating Fujitsu Enterprise Postgres memory requirements.

Use the following formula to obtain a rough estimate of memory required for Fujitsu Enterprise Postgres:

```
fujitsuEnterprisePostgresRequiredMemory = sharedMemoryAmount + localMemoryAmount
```

Shared memory amount

Refer to "Shared Memory and Semaphores" under "Server Administration" in the PostgreSQL Documentation for information on shared memory. If you enable the Global Meta Cache feature, you must also add the value of pgx_global_metacache. Refer to "Parameters" in the Operation Guide for the setting values.

However, note that if instances have been created using WebAdmin, the parameters below will be configured automatically when the instances are created. Take this into account when calculating the shared memory size.

Parameter name	Set value
shared_buffers	30 percent of the internal memory of the machine.
max_connections	100
max_prepared_transactions	100

Local memory amount

```
localMemoryAmount = processStackArea
                  + memoryUsedInDbSessionsThatUseTempTables
                  + memoryUsedInDbSessionsThatPerformSortAndHashTableOperations
                  + memoryUsedInMaintenanceOperations
                  + baseMemoryUsedInEachProcess
                  + memoryUsedPreparingForDataAccess
```

Process stack area

```
processStackArea
  = max_stack_depth x (max_connections + autovacuum_max_workers + 9)
```

This formula evaluates to the maximum value.
Actually it is used according to the growth of the stack.
In the formula above, 9 is the number of processes that perform roles specific to servers.

Memory used in database sessions that use temporary tables

```
memoryUsedInDbSessionsThatUseTempTables
  = temp_buffers x max_connections
```

This formula evaluates to the maximum value.
Memory is gradually used as temporary buffers are used, and is released when the session ends.

Memory used in database sessions that perform sort and hash table operations

```
memoryUsedInDbSessionsThatPerformSortAndHashTableOperations
  = work_mem (*1) x max_connections
```

*1) For hash table operations, multiply work_mem by hash_mem_multiplier.
This formula evaluates to the maximum value.
Memory is gradually used as operations such as sort are performed, and is released when the query ends.

Memory used in maintenance operations

```
memoryUsedInMaintenanceOperations
  = maintenance_work_mem x (numOfSessionsPerformingMaintenance + autovacuum_max_workers)
```

Note that 'maintenance operations' are operations such as VACUUM, CREATE INDEX, and ALTER TABLE ADD FOREIGN KEY.

Base memory used in each process

```
baseMemoryUsedInEachProcess
  = baseMemoryUsedInOneProcess x (max_connections + autovacuum_max_workers + 9)
```

Use the result of the following formula for memory consumed per process. This formula evaluates to the memory used when server processes are running.
In the formula above, 9 is the number of processes that perform roles specific to servers.

The amount of memory consumed per process is determined by the number of tables, indexes, and all columns of all tables that the process accesses. If your system has about 100 tables, you can estimate it to be 3 MB, but otherwise use the following estimate:

```
baseMemoryUsedInOneProcess
  = (1.9KB x All user tables + 2.9KB x All user indexes + 1.0KB x All user columns) x 1.5(*1)

If you enable the Global Meta Cache feature, use the following formula:
baseMemoryUsedInOneProcess
 = (All user tables + All user indexes + All user columns) x 1.0KB x 1.5 (*1)
    + (All user tables x 1.4KB + All user indexes x 2.4KB)
```

*1) Safety Factor (1.5)
There are variable length information. This value takes that into account.

Memory used preparing for data access

```
memoryUsedPreparingForDataAccess
 = variationAmount x (max_connections + autovacuum_max_workers + 4)

  where variationAmount = shared_buffers / 8KB x 4 bytes
      (note that 8KB is the page length, and 4 bytes is the size of page management data)
```

This formula evaluates to the memory required to access the database cache in the shared memory.
In the formula above, among the processes that perform roles specific to servers, 4 is the number of processes that access the database.

F.2 Database Multiplexing Memory Requirements

This section describes the formula for estimating database multiplexing memory requirements for the database server.

Use the following formula to obtain a rough estimate of memory required for database multiplexing:

```
Memory usage of the database multiplexing feature for the database server
           = Peak memory usage of the Mirroring Controller processes
           + Peak memory usage of the Mirroring Controller commands

Peak memory usage of the Mirroring Controller processes=150 MB

Peak memory usage of the Mirroring Controller commands=50 MB x Number of commands executed
simultaneously
```

F.3 VCI Memory Requirements

This section describes the formula for estimating VCI memory requirements.

Use the following formula to obtain a rough estimate of memory requirements:

```
memUsedByVci = memForData + memForEachProcess
```

Memory required to store data in memory

Secure the space estimated using the formula below on the stable buffer (part of shared_buffers).

```
memForData = (numOfRowsInTables) x (numOfBytesPerRow) + (wosSize)
```

Number of bytes per row

```
numOfBytesPerRow
 = (19 + (numOfColsInCreateIndexStatement) / 8 + (numOfBytesPerSingleColValue)) x 1.1
```

Note: Round up the result to the nearest integer.

WOS size

```
wosSize = (numOfWosRows) / 185 x 8096
```

One row is added to the number of WOS rows for each INSERT and DELETE, and two rows are added for UPDATE. On the other hand, the number decreases to 520,000 rows or less during conversion to ROS performed by the ROS control daemon.

Memory required for each process

```
memForEachProcess
 = memUsedPerScanning
 + memUsedForVciMaintenace
 + memUsedByCreateIndexStatement
```

Memory used per scanning

- Parallel scan

```
memUsedPerScanning
 = vci.shared_work_mem + (numOfParallelWorkers + 1) x vci.maintenance_work_mem
```

Note: The number of parallel workers used by VCI simultaneously in the entire instance is equal to or less than vci.max_parallel_degree.

- Non-parallel scan

```
memUsedPerScanning = vci.max_local_ros + vci.maintenance_work_mem
```

Note

- vci.shared_work_mem, and vci.max_local_ros are used to create local ROS. If local ROS exceeds these sizes, execute a query without using VCI according to the conventional plan.
- vci.maintenance_work_mem specifies the memory size to be secured dynamically. If it exceeds the specified value, a disk temporary file is used for operation.

Memory used for VCI maintenance

```
memUsedForVciMaintenace = vci.maintenance_work_mem x vci.control_max_workers
```

Memory used by CREATE INDEX

```
memUsedByCreateIndexStatement = vci.maintenance_work_mem
```

Note

vci.maintenance_work_mem specifies the memory to be secured dynamically. If it exceeds the specified value, a disk temporary file is used for operation.

F.4 High-Speed Data Load Memory Requirements

This section describes the formula for estimating memory requirements for the high-speed data load feature.

Use the following formula to obtain a rough estimate of memory requirements:

```
Memory usage of high speed data load
 = (Peak memory usage of pgx_loader processes + Peak memory usage of the pgx_loader commands)
 x Number of commands executed simultaneously

Peak memory usage of pgx_loader processes
 = Peak memory usage of the backend process   (6 MB)
 + Peak memory usage of parallel workers       (6 MB x number of parallel workers)
 + Peak memory usage of dynamic shared memory (80 MB x number of parallel workers)

Peak memory usage of the pgx_loader commands=9 MB
```

Point

In addition to the size calculated using the formula above, the database cache on the shared memory estimated using the shared_buffers parameter is consumed according to the size of the data (table and index keys) loaded using this feature. Refer to "E.1 Estimating Table Size Requirements" and "E.2 Estimating Index Size Requirements" for information on estimating an appropriate shared buffers value.

F.5 Global Meta Cache Memory Requirements

This section describes the formula for estimating Global Meta Cache memory requirements.

The memory calculated by "Size of the GMC area" is allocated to the shared memory. The memory calculated by the per-process meta cache management information is allocated to the local memory. Refer to the graphic in "Architecture of Global Meta Cache Feature" in the "Memory usage reduction by Global Meta Cache" in the General Description for more information.

Use the following formula to obtain a rough estimate of memory requirements:

```
Amount of memory used by the Global Meta Cache feature
  = Size of GMC area + Per-process meta cache management information

Size of GMC area = (All user tables x 0.4 KB
     + All user indexes x 0.3 KB
     + All user columns x 0.8 KB) x 1.5 (*1)

Per-process meta cache management information
  = (All user tables + All user indexes + All user columns) x 0.1KB x max_connections x 1.5 (*1)
```

*1) Safety Factor (1.5)
This value takes into account the case where both GMC before and after the change temporarily exist at the same time in shared memory when the table definition is changed or the row of the system catalog is changed.

Appendix G Quantitative Limits

This appendix lists the quantitative limits of Fujitsu Enterprise Postgres.

Table G.1 Length of identifier

Item	Limit
Database name	Up to 63 bytes (*1) (*2)
Schema name	Up to 63 bytes (*1) (*2)
Table name	Up to 63 bytes (*1) (*2)
View name	Up to 63 bytes (*1) (*2)
Index name	Up to 63 bytes (*1) (*2)
Tablespace name	Up to 63 bytes (*1) (*2)
Cursor name	Up to 63 bytes (*1) (*2)
Function name	Up to 63 bytes (*1) (*2)
Aggregate function name	Up to 63 bytes (*1) (*2)
Trigger name	Up to 63 bytes (*1) (*2)
Constraint name	Up to 63 bytes (*1) (*2)
Conversion name	Up to 63 bytes (*1) (*2)
Role name	Up to 63 bytes (*1) (*2)
Cast name	Up to 63 bytes (*1) (*2)
Collation sequence name	Up to 63 bytes (*1) (*2)
Encoding method conversion name	Up to 63 bytes (*1) (*2)
Domain name	Up to 63 bytes (*1) (*2)
Extension name	Up to 63 bytes (*1) (*2)
Operator name	Up to 63 bytes (*1) (*2)
Operator class name	Up to 63 bytes (*1) (*2)
Operator family name	Up to 63 bytes (*1) (*2)
Rewrite rule name	Up to 63 bytes (*1) (*2)
Sequence name	Up to 63 bytes (*1) (*2)
Text search settings name	Up to 63 bytes (*1) (*2)
Text search dictionary name	Up to 63 bytes (*1) (*2)
Text search parser name	Up to 63 bytes (*1) (*2)
Text search template name	Up to 63 bytes (*1) (*2)
Data type name	Up to 63 bytes (*1) (*2)
Enumerator type label	Up to 63 bytes (*1) (*2)

*1: This is the character string byte length when converted by the server character set character code.

*2: If an identifier that exceeds 63 bytes in length is specified, the excess characters are truncated and it is processed.

Table G.2 Database object

Item	Limit
Number of databases	Less than 4,294,967,296 (*1)

Item	Limit
Number of schemas	Less than 4,294,967,296 (*1)
Number of tables	Less than 4,294,967,296 (*1)
Number of views	Less than 4,294,967,296 (*1)
Number of indexes	Less than 4,294,967,296 (*1)
Number of tablespaces	Less than 4,294,967,296 (*1)
Number of functions	Less than 4,294,967,296 (*1)
Number of aggregate functions	Less than 4,294,967,296 (*1)
Number of triggers	Less than 4,294,967,296 (*1)
Number of constraints	Less than 4,294,967,296 (*1)
Number of conversion	Less than 4,294,967,296 (*1)
Number of roles	Less than 4,294,967,296 (*1)
Number of casts	Less than 4,294,967,296 (*1)
Number of collation sequences	Less than 4,294,967,296 (*1)
Number of encoding method conversions	Less than 4,294,967,296 (*1)
Number of domains	Less than 4,294,967,296 (*1)
Number of extensions	Less than 4,294,967,296 (*1)
Number of operators	Less than 4,294,967,296 (*1)
Number of operator classes	Less than 4,294,967,296 (*1)
Number of operator families	Less than 4,294,967,296 (*1)
Number of rewrite rules	Less than 4,294,967,296 (*1)
Number of sequences	Less than 4,294,967,296 (*1)
Number of text search settings	Less than 4,294,967,296 (*1)
Number of text search dictionaries	Less than 4,294,967,296 (*1)
Number of text search parsers	Less than 4,294,967,296 (*1)
Number of text search templates	Less than 4,294,967,296 (*1)
Number of data types	Less than 4,294,967,296 (*1)
Number of enumerator type labels	Less than 4,294,967,296 (*1)
Number of default access privileges defined in the ALTER DEFAULT PRIVILEGES statement	Less than 4,294,967,296 (*1)
Number of large objects	Less than 4,294,967,296 (*1)
Number of index access methods	Less than 4,294,967,296 (*1)

*1: The total number of all database objects must be less than 4,294,967,296.

Table G.3 Schema element

Item	Limit
Number of columns that can be defined in one table	From 250 to 1600 (according to the data type)
Table row length	Up to 400 gigabytes
Number of columns comprising a unique constraint	Up to 32 columns
Data length comprising a unique constraint	Less than 2,000 bytes (*1) (*2)

Item	Limit
Table size	Up to 32 terabyte
Search condition character string length in a trigger definition statement	Up to 800 megabytes (*1) (*2)
Item size	Up to 1 gigabyte

*1: Operation might proceed correctly even if operations are performed with a quantity outside the limits.

*2: This is the character string byte length when converted by the server character set character code.

Table G.4 Index

Item	Limit
Number of columns comprising a key (including VCI)	Up to 32 columns
Key length (other than VCI)	Less than 2,000 bytes (*1)

*1: This is the character string byte length when converted by the server character set character code.

Table G.5 Data types and attributes that can be handled

Item			Limit
Character	Data length		Data types and attributes that can be handled (*1)
	Specification length (n)		Up to 10,485,760 characters (*1)
Numeric	External decimal expression		Up to 131,072 digits before the decimal point, and up to 16,383 digits after the decimal point
	Internal binary expression	2 bytes	From -32,768 to 32,767
		4 bytes	From -2,147,483,648 to 2,147,483,647
		8 bytes	From -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
	Internal decimal expression		Up to 13,1072 digits before the decimal point, and up to 16,383 digits after the decimal point
	Floating point expression	4 bytes	From -3.4E+38 to -7.1E-46, 0, or from 7.1E-46 to 3.4E+38
		8 bytes	From -1.7E+308 to -2.5E-324, 0, or from 2.5E-324 to 1.7E+308
bytea			Up to one gigabyte minus 53 bytes
Large object			Up to 4 terabyte

*1: This is the character string byte length when converted by the server character set character code.

Table G.6 Function definition

Item	Limit
Number of arguments that can be specified	Up to 100
Number of variable names that can be specified in the declarations section	No limit
Number of SQL statements or control statements that can be specified in a function processing implementation	No limit

Table G.7 Data operation statement

Item	Limit
Maximum number of connections for one process in an application (remote access)	4,000 connections
Number of expressions that can be specified in a selection list	Up to 1,664
Number of tables that can be specified in a FROM clause	No limit
Number of unique expressions that can be specified in a selection list/ DISTINCT clause/ORDER BY clause/GROUP BY clause within one SELECT statement	Up to 1,664
Number of expressions that can be specified in a GROUP BY clause	No limit
Number of expressions that can be specified in an ORDER BY clause	No limit
Number of SELECT statements that can be specified in a UNION clause/INTERSECT clause/EXCEPT clause	Up to 4,000 (*1)
Number of nestings in joined tables that can be specified in one view	Up to 4,000 (*1)
Number of functions or operator expressions that can be specified in one expression	Up to 4,000 (*1)
Number of expressions that can be specified in one row constructor	Up to 1,664
Number of expressions that can be specified in an UPDATE statement SET clause	Up to 1,664
Number of expressions that can be specified in one row of a VALUES list	Up to 1,664
Number of expressions that can be specified in a RETURNING clause	Up to 1,664
Total expression length that can be specified in the argument list of one function specification	Up to 800 megabytes (*2)
Number of cursors that can be processed simultaneously by one session	No limit
Character string length of one SQL statement	Up to 800 megabytes (*1) (*3)
Number of input parameter specifications that can be specified in one dynamic SQL statement	No limit
Number of tokens that can be specified in one SQL statement	Up to 10,000
Number of values that can be specified as a list in a WHERE clause IN syntax	No limit
Number of expressions that can be specified in a USING clause	No limit
Number of JOINs that can be specified in a joined table	Up to 4,000 (*1)
Number of expressions that can be specified in COALESCE	No limit
Number of WHEN clauses that can be specified for CASE in a simple format or a searched format	No limit
Data size per record that can be updated or inserted by one SQL statement	Up to one gigabyte minus 53 bytes
Number of objects that can share a lock simultaneously	Up to 256,000 (*1)

*1: Operation might proceed correctly even if operations are performed with a quantity outside the limits.

*2: The total number of all database objects must be less than 4,294,967,296.

*3: This is the character string byte length when converted by the server character set character code.

Table G.8 Data size

Item	Limit
Data size per record for input data files (COPY statement, psql command \copy meta command)	Up to 800 megabytes (*1)
Data size per record for output data files (COPY statement, psql command \copy meta command)	Up to 800 megabytes (*1)

*1: Operation might proceed correctly even if operations are performed with a quantity outside the limits.

Appendix H Configuring Kernel Parameters

Use the "System V IPC Parameters" table in "Managing Kernel Resources" in the PostgreSQL Documentation for the relationship between configuration parameters and kernel parameters, as well as calculation formulas.
Refer to the "Managing Kernel Resources" in the PostgreSQL Documentation to calculate shared memory usage.

For multiple instances, the kernel parameters should be evaluated for all instances. For example, in the case of the maximum number of shared memory segments for the entire system (SHMMNI), the total number of segments obtained by all instances should be added to the kernel parameters. In the case of the maximum number of semaphores for each process (SEMMSL), the largest of all sizes obtained by all instances should be compared to the current value prior to configuring the settings.

Note

If there is insufficient shared memory due to miscalculation of SHMMAX, a message will be output indicating that the shmget system call failed at "errno=22 (EINVAL)". Review the calculation, and reconfigure.

The relationship between System V IPC parameters and kernel parameters in various operating systems is shown below.

System	V IPC parameter	Kernel parameter action
SHMMAX	kernel.shmmax	If *currentValue* < *calculatedValue*, configure the calculated value
SHMMIN	No compatible parameter	
SHMALL	kernel.shmall	Specify *currentValue* + *calculatedValue*
SHMSEG	No compatible parameter	
SHMMNI	kernel.shmmni	Specify *currentValue* + *calculatedValue*
SEMMNI	Fourth parameter of kernel.sem	Specify *currentValue* + *calculatedValue*
SEMMNS	Second parameter of kernel.sem	Specify *currentValue* + *calculatedValue*
SEMMSL	First parameter of kernel.sem	If *currentValue* < *calculatedValue*, configure the calculated value
SEMMAP	No compatible parameter	
SEMVMX	No compatible parameter	

Remark 1: kernel.shmall specifies the number of pages.
Remark 2: Specify all four parameters for kernel.sem. At this time, the value specified in the third parameter should be the same value as before configuration.

Appendix I Determining the Preferred WebAdmin Configuration

This appendix describes the two different configurations in which WebAdmin can be used and how to select the most suitable configuration.

I.1 WebAdmin Configurations

WebAdmin can be installed in two configurations:

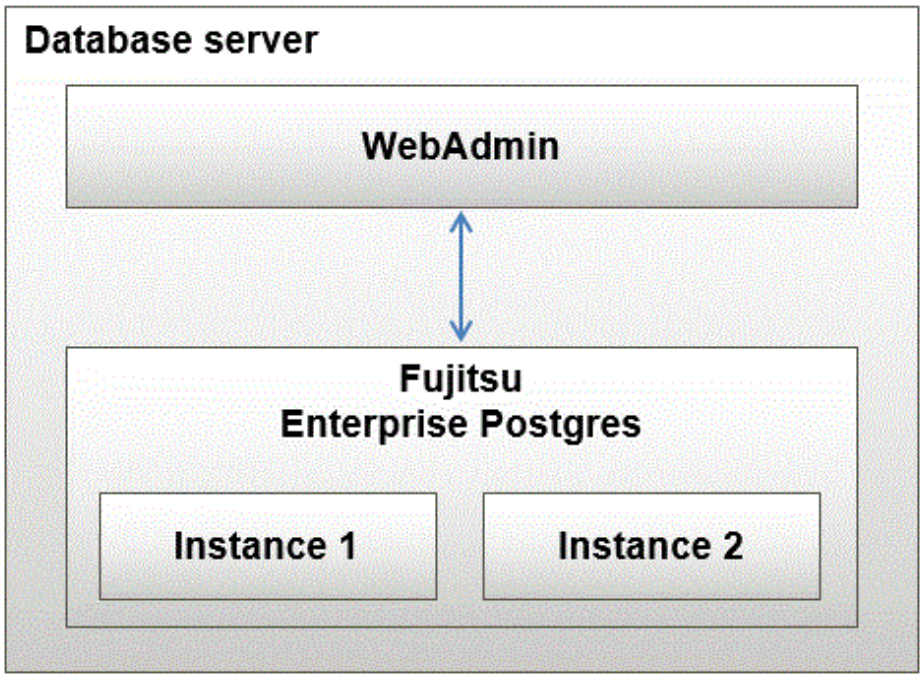
- Single-server
- Multiserver

WebAdmin does not support encrypted communication between browser and server or between servers. Therefore, when using WebAdmin in either configuration, build the communication path with the browser or each server on a network that cannot be accessed externally.

I.1.1 Single-Server Configuration

A single-server configuration enables you to create and operate instances on a single server. In this configuration, WebAdmin must be installed on the same database server as the Fujitsu Enterprise Postgres Server component.

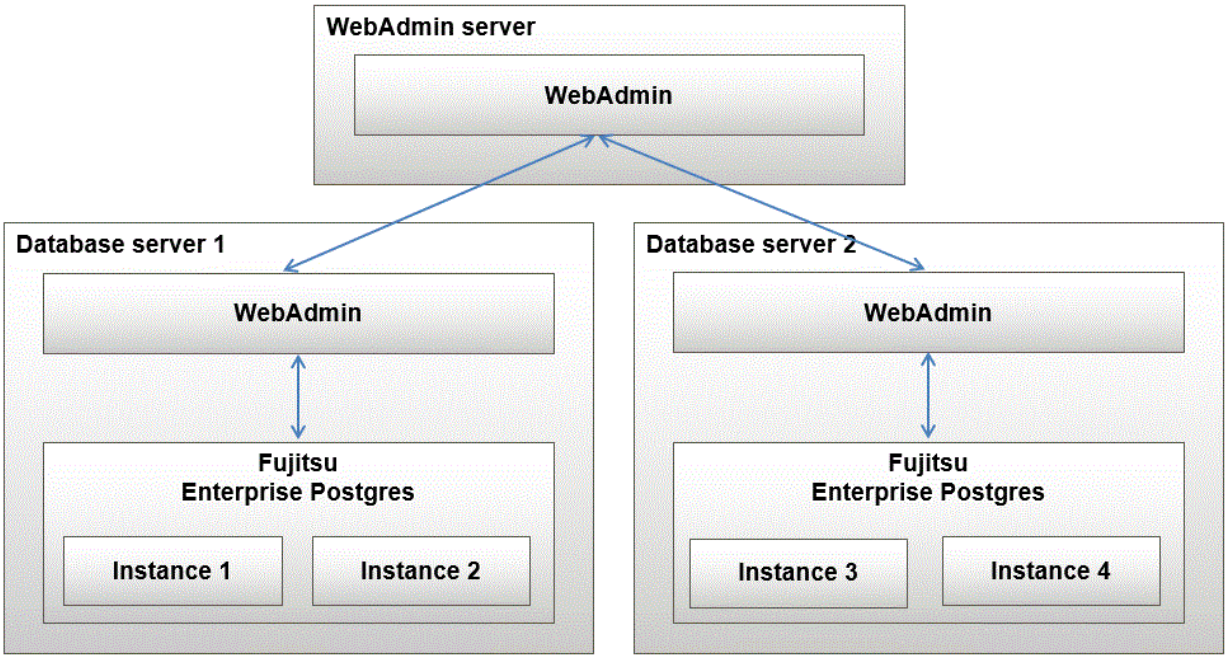
Single-server configuration



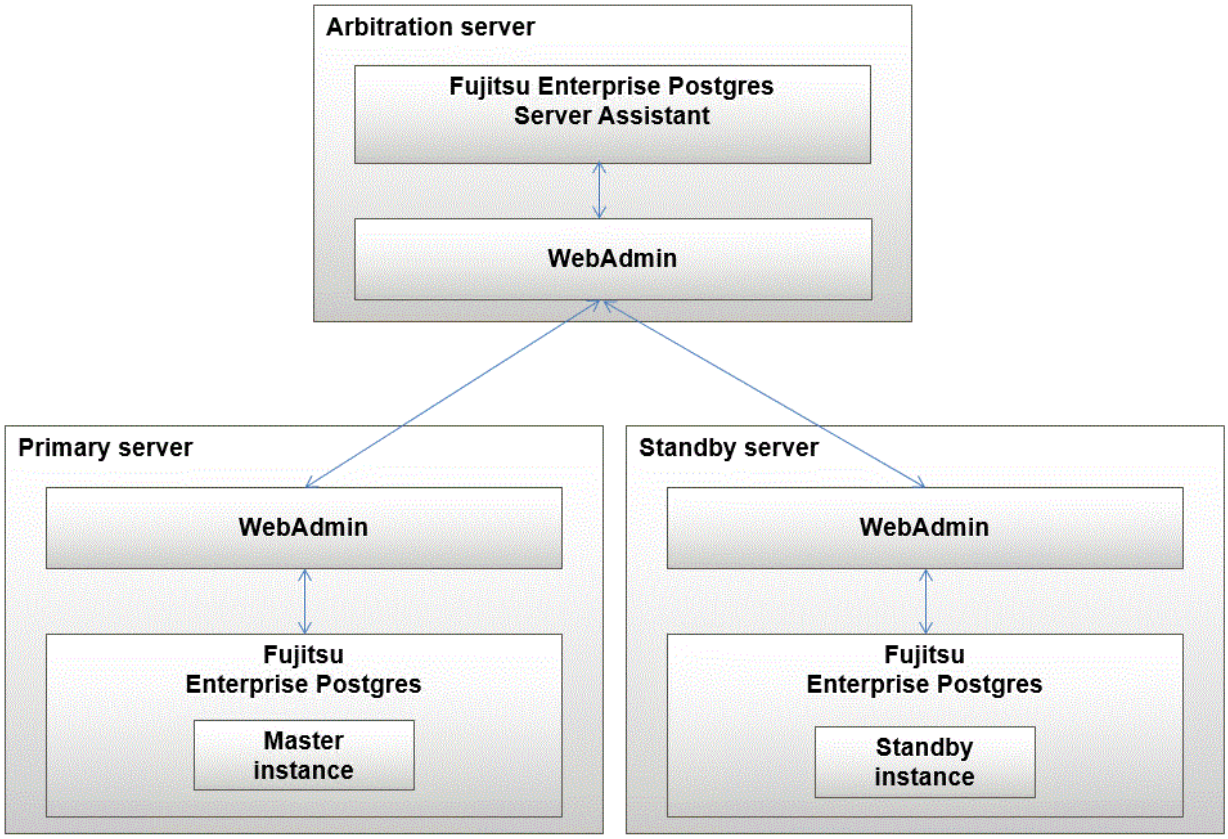
I.1.2 Multiserver Configuration

A multiserver configuration enables you to create and operate instances stored on multiple database servers. As shown in the figure below, WebAdmin can be installed on a dedicated WebAdmin server and used to collectively manage the instances stored on the database servers.

Multiserver configuration



Also, when setting up the arbitration server by WebAdmin during database multiplexing mode, install WebAdmin on the arbitration server.



I.2 Installing WebAdmin in a Single-Server Configuration

To install WebAdmin in a single-server configuration, the Fujitsu Enterprise Postgres Server component and WebAdmin must be installed on the same machine.

Select the following items when installing Fujitsu Enterprise Postgres in a single-server configuration:

- Fujitsu Enterprise Postgres Advanced Edition or Fujitsu Enterprise Postgres Standard Edition
- WebAdmin

I.3 Installing WebAdmin in a Multiserver Configuration

In a multiserver configuration, install WebAdmin on one server, and both WebAdmin and the Fujitsu Enterprise Postgres Server component on any number of database servers.

Select the following items when installing Fujitsu Enterprise Postgres in a multiserver configuration:

- WebAdmin server:
  - WebAdmin
- Database server:
  - Fujitsu Enterprise Postgres Advanced Edition or Fujitsu Enterprise Postgres Standard Edition
  - WebAdmin

Also, when setting up the arbitration server by WebAdmin during database multiplexing mode, select the following when installing Fujitsu Enterprise Postgres.

- Arbitration server
  - Fujitsu Enterprise Postgres Server Assistant
  - WebAdmin

See

Refer to the Installation and Setup Guide for Server Assistant for details on how to install the Server Assistant.

Appendix J System Configuration when using Pgpool-II

Describes the system configuration when using Pgpool-II.

The system configuration when using Pgpool-II is as follows:

Place on database server

System configuration to coexist the database server with Pgpool-II.

Place on application server

System configuration to coexist the application server with Pgpool-II.

Place on dedicated server

System configuration in which Pgpool-II resides on a dedicated server (Pgpool-II Server) that is separate from the database and application servers.

Select the system configuration that best meets your operational requirements.

J.1 Pgpool-II Configuration

In this example, Pgpool-II is deployed on a different Pgpool-II server than the database and application servers.

There are three configurations of Pgpool-II:

- Single-machine configuration
- Two-machine configuration
- Three-machine configuration

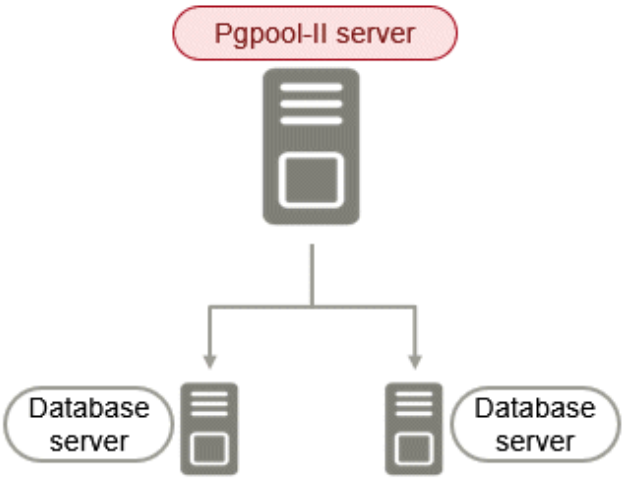
Although the Pgpool-II server can be operated on a single machine, to ensure business continuity, it is recommended to operate the Pgpool-II server using a three-machine configuration in Fujitsu Enterprise Postgres.

If employing a configuration of three or more machines, use an odd number of machines in the configuration.

J.1.1 Single-Machine Configuration

This is the basic configuration when running Pgpool-II.

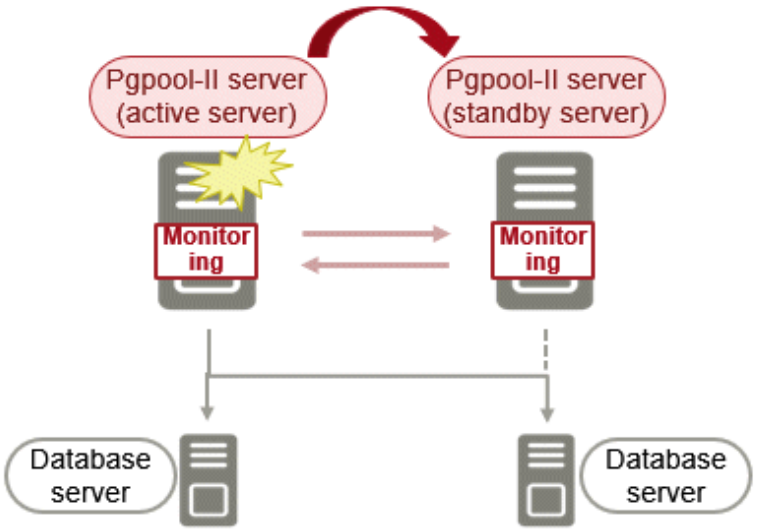
Although the database server has redundancy, if an error occurs on the Pgpool-II server that accesses the database server, the job will stop.



J.1.2 Two-Machine Configuration

When an error occurs on the active server, the Pgpool-II monitoring feature that mutually monitors the status of the Pgpool-II servers enables jobs to continue uninterrupted by switching to the standby server.

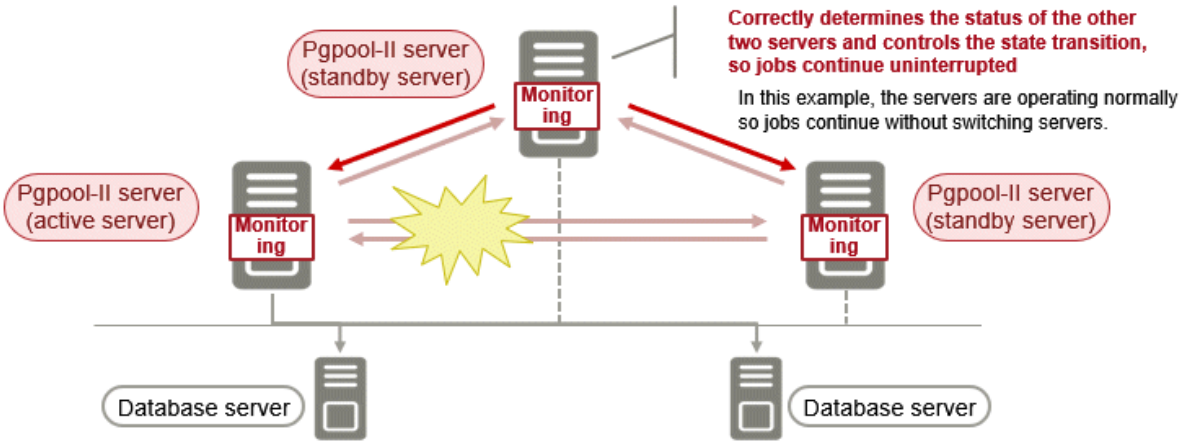
If the network between Pgpool-II servers is disconnected, even if the Pgpool-II servers are running correctly, which may lead to stoppage of jobs.



J.1.3 Three-Machine Configuration

The Pgpool-II monitoring feature enables a Pgpool-II server to monitor the other two Pgpool-II servers.

Even if any of the networks monitoring the Pgpool-II servers are disconnected, the status of servers on a network that is operating normally can be checked correctly, enabling accurate continuation of jobs.



J.2 Installing Pgpool-II

Pgpool-II is bundled with the server program and the client program. To use Pgpool-II, use the server program or the client program to install and set up Pgpool-II.

Depending on where Pgpool-II is installed, select the appropriate DVD for deployment:

Installing on Database Server (coexist)

Install the Pgpool-II program along with the server program from the server program DVD.

Installing on Application Server (coexist)

Install the Pgpool-II program along with the client program from the client program DVD.

Installing on Dedicated server different from the above (Pgpool-II server)

Install the Pgpool-II program along with the client program from the client program DVD.

J.3 Pgpool-II Setup

Describes how to set up Pgpool-II.

J.3.1 Setting Environment Variables

If you use the Pgpool-II command, set the following environment variables:

PATH environment variable

Add "Install Directory/bin".

The following is an example of setting environment variables:

Example

The following is an example of setting environment variables when the installation directory is "/ opt/fsepv < x > pgpool-II".

"< x >"” indicates the product version.

```
$ PATH=/opt/fsepv<x>pgpool-II/bin:$PATH ; export PATH
```

J.3.2 Configuration file

Describes Pgpool-II configuration files.

J.3.2.1 Configuring pgpool.conf

To configure pgpool.conf, see the Pgpool-II documentation.

A sample configuration file is located under the installation directory/etc.

J.3.2.2 Using Configuration Files

The pgpool command makes use of configuration files such as pgpool.conf, pcp.conf, and pool_hba.conf.

To take advantage of these configuration files, specify the path to the files in the pgpool command options.

The following example shows how to configure options for the pgpool command:

Example

```
$ pgpool -f /usr/local/etc/pgpool.conf -F /usr/local/etc/pcp.conf -a / usr/local/etc/pool_hba.conf
```

Appendix K Supported contrib Modules and Extensions Provided by External Projects

Fujitsu Enterprise Postgres supports PostgreSQL contrib modules, and extensions provided by external projects.

Refer to the following for details on the supported contrib modules:

- "Additional Supplied Modules" in the PostgreSQL Documentation
- "Additional Supplied Programs" in the PostgreSQL Documentation

Information

You can also check the list of available extensions using the pg_available_extensions view.

Refer to "OSS Supported by Fujitsu Enterprise Postgres" in the General Description for information on supported extensions provided by external projects.

Appendix L Procedure when Modifying the JRE Installation

This appendix describes the procedures to follow when modifying the JRE installation.

The JRE, of which the installation destination is specified in the JAVA_HOME environment variable when installing Fujitsu Enterprise Postgres, is used by features such as WebAdmin and database multiplexing.

Therefore, when updating or reinstalling JRE after installing Fujitsu Enterprise Postgres, the procedures below must be performed.

L.1 When Using WebAdmin

WebAdmin must be set up again.

Follow the procedure below to modify the JRE installation:

1. Stop the Web server feature of WebAdmin

   Refer to "B.1.3 Stopping the Web Server Feature of WebAdmin" for details.

2. Remove WebAdmin

   Refer to "B.2 Removing WebAdmin" for details.

3. Modify the JRE installation

4. Set the JAVA_HOME environment variable

   Set the JAVA_HOME environment variable to the installation destination of JRE 8.

   Example

   ```
   # export JAVA_HOME="Jre8InstallDir"
   ```

5. Set up WebAdmin

   Refer to "B.1.1 Setting Up WebAdmin" for details.

6. Start the Web server feature of WebAdmin

   Refer to "B.1.2 Starting the Web Server Feature of WebAdmin" for details.

L.2 When Performing Database Multiplexing

Mirroring Controller must be restarted.

Follow the procedure below to modify the JRE installation:

1. Stop Mirroring Controller

   Refer to the Cluster Operation Guide (Database Multiplexing) for details.

2. Modify the JRE installation

3. Change the installation environment to be used by Mirroring Controller

   If database multiplexing is performed using WebAdmin, perform the procedure described in this procedure after performing step 4 "Set the JAVA_HOME environment variable" in "L.1 When Using WebAdmin".

   Set the JAVA_HOME environment variable to the installation destination of JRE 8, and use the mc_update_jre_env command to change the installation environment to be used by Mirroring Controller.
   This procedure must be executed by the superuser.

Example

/opt/fsepv<*x*>server64/bin is the installation directory where the server product is installed.

```
$ su -
Password:******
# export JAVA_HOME="Jre8InstallDir"
# /opt/fsepv<x>server64/bin/mc_update_jre_env
```

4. Start Mirroring Controller

   Refer to the Cluster Operation Guide (Database Multiplexing) for details.

Appendix M Access to Key Management System Using Plug-in

M.1 What to do with Plug-ins

Plug-ins are called to verify, encrypt, and decrypt keys.

Key validation, encryption, and decryption requests are required.

M.2 Where the Plug-in is Stored

Plug-ins are stored as executables with the same name as the plug-in name in the directory specified in the tde_kms.plugin.path parameter. It is the responsibility of the database administrator to ensure that only secure plugins are stored in this directory.

M.3 Invoking the Plug-in

The plug-in runs with the same ownership as the user running the FEP server. The plug-in is passed information that should be kept secret. It is the database administrator's responsibility to ensure that the plug-in is trustworthy.

Plug-ins can be invoked at the same time (multiple).

The plug-in must complete the operation in a timely manner and return a response.

M.4 Passing Confidential Information to Plug-ins

Confidential credentials passed to the FEP instance using the FEP keystore open facility (pgx _ open _ keystore function, opening at server startup prompt, opening using obfuscated files) are passed to the plug-in as environment variables.

You can pass arbitrary values as arguments when calling the plug-in, but do not use this feature to pass sensitive information.

The authentication and authorization of access to the key management system depends on the implementation of the plug-in.

M.5 Calling Convention

M.5.1 Key Verification

Arguments

The following arguments are supplied:

	Argument value	Notes
First argument	validate-key	Fixed
second argument	--keyid	Fixed
third argument	*keyid*	Variable; the key ID specified in the pgx_declare_external_master_key function is passed
After the fourth argument	*extraarg*	Arguments specified in the connection information file, if any, are given in the specified order

Environment variable

The following environment variables are supplied:

Name of the environment variable	Value of the environment variable	Notes
TDE_KMS_SECRET	KMS Secret	String entered in the FEP to open the keystore

Return value

The command ends with the following return values:

Return value	Condition
0	If the processing is successful
Other than 0	When processing does not complete normally

delivery of data

Data is delivered to the plug-in in the following way.The plug-in also returns results in the following ways:

Classification	Data Content	Delivery method	Notes
Input	Key ID	Arguments	
Output	Process Status	Plug-in return code	
Output	Message	Plugin standard error output	Expected to be printable

Calling opportunity

Called before starting to use the encryption key.

Processing contents

Verifies the existence of the encryption key identified by the key ID and whether the user is authorized to use the encryption key.

M.5.2 Encryption

Arguments

The following arguments are supplied:

	Argument value	Notes
First argument	encrypt	Fixed
second argument	--keyid	Fixed
third argument	*keyid*	Variable; the key ID specified in the pgx_declare_external_master_key function is passed
After the fourth argument	*extraarg*	The values specified in extra-args in the key management system connection information file, if any, are passed in the specified order

Environment variable

The following environment variables are supplied:

Name of the environment variable	Value of the environment variable	Notes
TDE_KMS_SECRET	KMS Secret	String entered in the FEP to open the keystore

Return value

The command ends with the following return values:

Return value	Condition
0	If the processing is successful
Other than 0	When processing does not complete normally

delivery of data

Data is delivered to the plug-in in the following way.The plug-in also returns results in the following ways:

Classification	Data Content	Delivery method	Notes
Input	Data to be encrypted	Standard input for the plug-in	As Is (not Base 64 encoding, etc)
Input	Key ID	Arguments	
Input	Encryption parameter	-	Not passed by the FEP
Output	Encryption result	Standard output of the plug-in	As Is (not Base 64 encoding, etc)
Output	Status of the action	plug-in return code	
Output	Message	Standard error output of the plug-in	It is expected to be printable

Calling opportunity

Called when encryption with the master encryption key is required.

Processing Contents

Encrypts the given data to be encrypted with the encryption key identified by the specified key ID, and returns the result. The returned encryption result must be decryptable with the same key ID.

Caution

- Implement so that data to be encrypted is not leaked. For example, temporarily storing encrypted data that is plaintext in a file poses a risk of disclosure.
- Fujitsu Enterprise Postgres only guarantees the following during decryption:
  - The same key ID is handed over during decryption as during encryption.
  - The data received as a result of the "encryption" operation is passed as-is when decrypting.
- The maximum amount of data to be encrypted passed from the Fujitsu Enterprise Postgres is 2048 bytes.

M.5.3 Decryption

Arguments

The following arguments are supplied:

	Argument value	Notes
First argument	decrypt	Fixed
second argument	--keyid	Fixed
third argument	*keyid*	Variable; the key ID specified in the pgx_declare_external_master_key function is passed
After the fourth argument	*extraarg*	The values specified in extra-args in the key management system connection information file, if any, are passed in the specified order.

Environment variable

The following environment variables are supplied:

Name of the environment variable	Value of the environment variable	Notes
TDE_KMS_SECRET	KMS Secret	String entered in the FEP to open the keystore

Return value

The command ends with the following return values:

Return value	Condition
0	If the processing is successful
Other than 0	When processing does not complete normally

delivery of data

Data is delivered to the plug-in in the following way.The plug-in also returns results in the following ways:

Classification	Data Content	Delivery method	Notes
Input	Data to be decrypted	Standard input for the plug-in	As Is (not Base 64 encoding, etc.)
Input	Key ID	Arguments	
Input	Encryption parameter	-	Not passed by the FEP
Output	Decoding result	Standard output of the plug-in	As Is (not Base 64 encoding, etc.)
Output	Status of the action	plug-in return code	
Output	Message	Standard error output of the plug-in	It is expected to be printable

Calling opportunity

Called when decryption with the master encryption key is required.

Processing Contents

Decrypts the given encrypted data with the encryption key identified by the given key ID and returns the result.

Caution

- Implement so that the decrypted data is not leaked. For example, there is a risk of leakage if the decryption result data, which is clear text, is temporarily stored in a file.
- If decryption requires the same encryption parameters as encryption, it is the plug-in's responsibility to ensure this. During decryption, the FEP only ensures that the plug-in receives the same key ID and encrypted data as was encrypted.
- The data to be decrypted is passed as is the data returned by the plug-in in response to the encryption request.

Index

[A]

Access to key management system using plug-in..... 91

[C]

Changing client authentication information..... 31
Changing Instance Settings..... 30
Check the disk space.....14
Check the installed products and determine the installation method..... 14
Client Authentication Information settings..... 36
Creating an Instance..... 27,33
Creating an Instance Administrator.....21
Creating Instances.....25

[D]

Disk Space Required for Installation.....11

[E]

Editing instance information..... 31
Excluded Software.....10

[H]

Hardware Environment.....11
How to Set Up the Pop-up Blocker..... 55

[I]

Importing Instances..... 32
Installation..... 14
Installation in Interactive Mode.....2
Installation in Silent Mode..... 2,18
Installation Procedure..... 1
Installation Types..... 1
Instance configuration..... 30

[L]

Logging in to WebAdmin.....26

[M]

Multi-Version Installation..... 1

[N]

New Installation.....1

[O]

Operating Environment..... 3
Operating Method Types and Selection..... 20

[P]

Port number to use when Tomcat is stopped.....56
postgresql.conf.....62
Pre-installation Tasks..... 14
Preparations for Setup..... 21
Procedure when Modifying the JRE Installation.....89

[R]

Recommended Browser Settings.....55
Reinstallation..... 1
Related Software.....9
Remove applied updates.....15
Removing WebAdmin.....58
Required Operating System.....3
Required Patches..... 11

[S]

Settings related to connection.....36
Starting the Web Server Feature of WebAdmin.....57
Startup URL for WebAdmin..... 26
Stopping the Web Server Feature of WebAdmin.....57
Supported contrib Modules and Extensions Provided by External Projects..... 88

[T]

TCP/IP Protocol.....12

[U]

Uninstallation.....2,51
Uninstallation in Interactive Mode.....51
Uninstallation in Silent Mode.....53
Using the initdb Command.....33
Using WebAdmin.....26

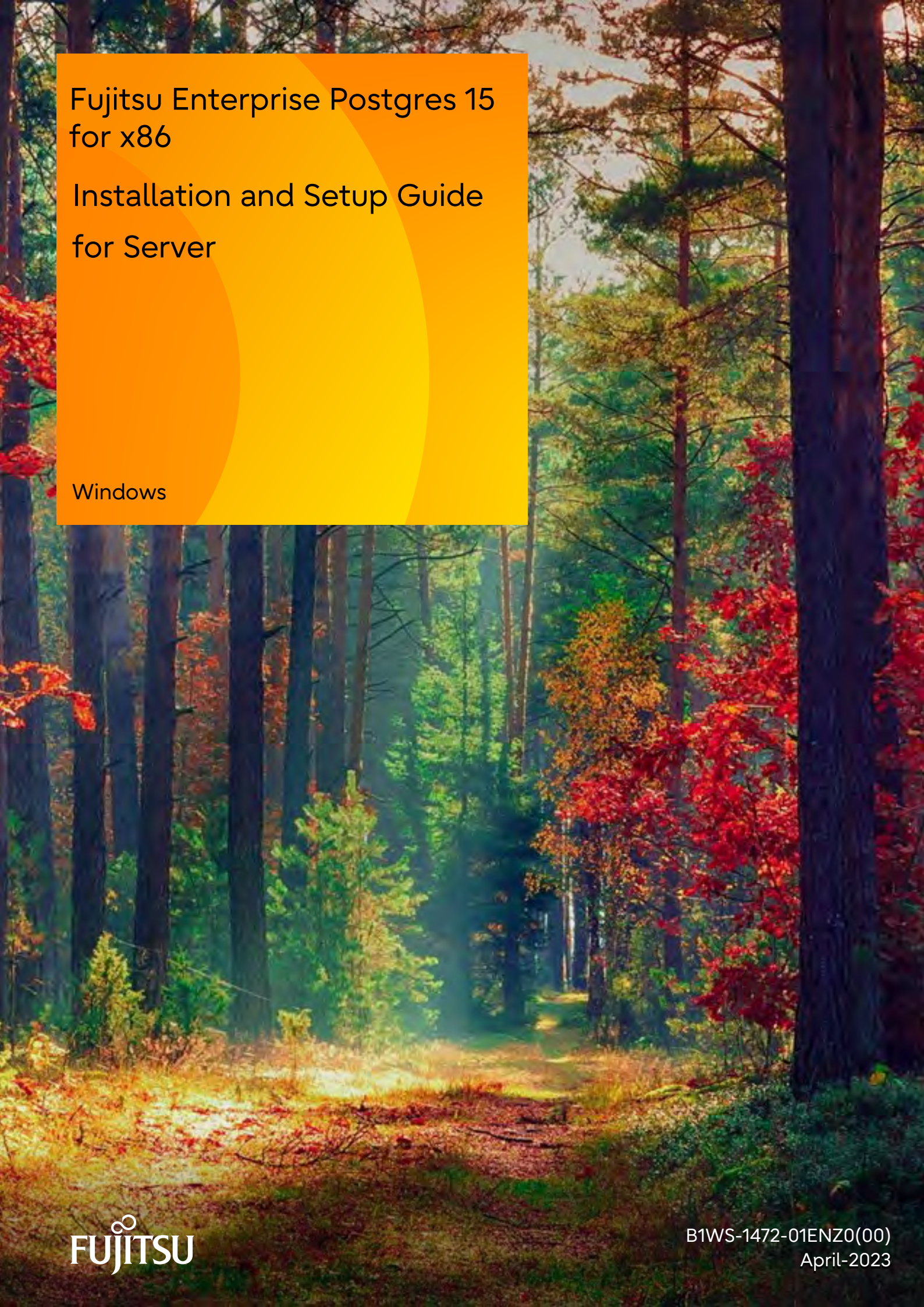
[W]

WebAdmin automatic start.....57
Web server port number..... 56
When an Instance was Created with the initdb Command.....36
When an Instance was Created with WebAdmin..... 36
When Performing Database Multiplexing.....89
When Using WebAdmin..... 89

Fujitsu Enterprise Postgres 15 for x86

Installation and Setup Guide for Server

Windows



FUJITSU

B1WS-1472-01ENZ0(00)
April-2023

Preface

Purpose of this document

The Fujitsu Enterprise Postgres database system extends the PostgreSQL features and runs on the Windows platform.

This document describes how to install and set up "Fujitsu Enterprise Postgres".

Intended readers

This document is intended for those who install and operate Fujitsu Enterprise Postgres.

Readers of this document are assumed to have general knowledge of:

- PostgreSQL
- SQL
- Windows

Structure of this document

This document is structured as follows:

Chapter 1 Overview of Installation

Describes the installation types and procedures

Chapter 2 Operating Environment

Describes the operating environment required to use Fujitsu Enterprise Postgres

Chapter 3 Installation

Describes how to perform a new installation of Fujitsu Enterprise Postgres

Chapter 4 Setup

Describes the setup to be performed after installation

Chapter 5 Uninstallation

Describes how to uninstall Fujitsu Enterprise Postgres

Appendix A Recommended WebAdmin Environments

Describes the recommended WebAdmin environment

Appendix B Setting Up and Removing WebAdmin

Describes how to set up and remove WebAdmin

Appendix C WebAdmin Disallow User Inputs Containing Hazardous Characters

Describes characters that are not allowed in WebAdmin.

Appendix D Configuring Parameters

Describes Fujitsu Enterprise Postgres parameters

Appendix E Estimating Database Disk Space Requirements

Describes how to estimate database disk space requirements

Appendix F Estimating Memory Requirements

Describes the formulas for estimating memory requirements

Appendix G Quantitative Limits

Describes the quantity range

Appendix H Determining the Preferred WebAdmin Configuration

Describes the two different configurations in which WebAdmin can be used and how to select the most suitable configuration

Appendix I Supported contrib Modules and Extensions Provided by External Projects

Lists the PostgreSQL contrib modules and the extensions provided by external projects supported by Fujitsu Enterprise Postgres.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

Edition 1.0: April 2023

Copyright

Copyright 2015-2023 Fujitsu Limited

Contents

Chapter 1 Overview of Installation....1
1.1 Features that can be Installed....1
1.2 Installation Types....1
1.2.1 New Installation....1
1.2.2 Reinstallation....1
1.2.3 Multi-Version Installation....1
1.3 Installation Procedure....1
1.3.1 Installation in Interactive Mode....2
1.3.2 Installation in Silent Mode....2
1.4 Uninstallation....2

Chapter 2 Operating Environment....3
2.1 Required Operating System....3
2.2 Related Software....3
2.3 Excluded Software....4
2.4 Required Patches....5
2.5 Hardware Environment....5
2.6 Disk Space Required for Installation....5
2.7 Supported System Environment....5
2.7.1 TCP/IP Protocol....5
2.7.2 File System....6
2.8 PostgreSQL Version Used for Fujitsu Enterprise Postgres....6
2.9 Notes on Using Streaming Replication....6

Chapter 3 Installation....7
3.1 Pre-installation Tasks....8
3.2 Installation in Interactive Mode....9
3.3 Installation in Silent Mode....11

Chapter 4 Setup....13
4.1 Operating Method Types and Selection....13
4.2 Preparations for Setup....14
4.2.1 Creating an Instance Administrator....14
4.2.1.1 Security policy settings....15
4.2.2 Preparing Directories for Resource Deployment....15
4.2.3 Estimating Resources....19
4.2.4 Windows Firewall Settings....19
4.2.5 Preparing for Output to the Event Log....20
4.3 Creating Instances....21
4.3.1 Using WebAdmin....22
4.3.1.1 Logging in to WebAdmin....22
4.3.1.2 Creating an Instance....23
4.3.1.3 Changing Instance Settings....26
4.3.1.4 Importing Instances....28
4.3.2 Using the initdb Command....30
4.3.2.1 Creating an Instance....30
4.4 Configuring Remote Connections....32
4.4.1 When an Instance was Created with WebAdmin....33
4.4.2 When an Instance was Created with the initdb Command....33
4.4.3 Windows Firewall Settings....33
4.5 Other Settings....34
4.5.1 Error Log Settings....34
4.5.2 Configuring Automatic Start and Stop of an Instance....35
4.5.3 Settings when Using the Features Compatible with Oracle Databases....36
4.6 Integration with Message-Monitoring Software....36
4.7 Setting Up and Removing OSS....36

4.7.1 oracle_fdw ... 37
4.7.1.1 Setting Up oracle_fdw ... 37
4.7.1.2 Removing oracle_fdw ... 37
4.7.2 pg_hint_plan ... 38
4.7.2.1 Setting Up pg_hint_plan ... 38
4.7.2.2 Removing pg_hint_plan ... 38
4.7.3 pg_dbms_stats ... 39
4.7.3.1 Setting Up pg_dbms_stats ... 39
4.7.3.2 Removing pg_dbms_stats ... 39
4.8 Deleting Instances ... 40
4.8.1 Using WebAdmin ... 40
4.8.2 Using Server Commands ... 40

Chapter 5 Uninstallation ... 42
5.1 Uninstallation in Interactive Mode ... 42
5.2 Uninstallation in Silent Mode ... 44

Appendix A Recommended WebAdmin Environments ... 48
A.1 Recommended Browser Settings ... 48
A.2 How to Set Up the Pop-up Blocker ... 48

Appendix B Setting Up and Removing WebAdmin ... 49
B.1 Setting Up WebAdmin ... 49
B.1.1 Setting Up WebAdmin ... 49
B.1.2 Starting the Web Server Feature of WebAdmin ... 50
B.1.3 Stopping the Web Server Feature of WebAdmin ... 50
B.2 Removing WebAdmin ... 51
B.3 Using an External Repository for WebAdmin ... 51
B.4 Using the WebAdmin Auto-Refresh Feature ... 53

Appendix C WebAdmin Disallow User Inputs Containing Hazardous Characters ... 54

Appendix D Configuring Parameters ... 55

Appendix E Estimating Database Disk Space Requirements ... 59
E.1 Estimating Table Size Requirements ... 59
E.2 Estimating Index Size Requirements ... 60
E.3 Sizes of Data Types ... 61
E.3.1 Sizes of Fixed-Length Data Types ... 61
E.3.2 Sizes of Variable-Length Data Types ... 62
E.3.3 Sizes of Array Data Types ... 62
E.3.4 Number of Bytes per Character ... 63
E.4 Estimating Transaction Log Space Requirements ... 63
E.5 Estimating Archive Log Space Requirements ... 63
E.6 Estimating Backup Disk Space Requirements ... 63
E.7 Estimating VCI Disk Space Requirements ... 63

Appendix F Estimating Memory Requirements ... 65
F.1 Fujitsu Enterprise Postgres Memory Requirements ... 65
F.2 Database Multiplexing Memory Requirements ... 66
F.3 VCI Memory Requirements ... 66
F.4 High-Speed Data Load Memory Requirements ... 68
F.5 Global Meta Cache Memory Requirements ... 68

Appendix G Quantitative Limits ... 69

Appendix H Determining the Preferred WebAdmin Configuration ... 74
H.1 WebAdmin Configurations ... 74
H.1.1 Single-Server Configuration ... 74

H.1.2 Multiserver Configuration 74
H.2 Installing WebAdmin in a Single-Server Configuration 75
H.3 Installing WebAdmin in a Multiserver Configuration 76

Appendix I Supported contrib Modules and Extensions Provided by External Projects 77

Index 78

Chapter 1 Overview of Installation

This chapter provides an overview of Fujitsu Enterprise Postgres installation.

1.1 Features that can be Installed

Each Fujitsu Enterprise Postgres feature is installed on the machine that was used to build the database environment.

The following table shows the relationship between the product to be installed and the features that can be installed.

Feature that can be installed	Product name	
	AE	SE
Basic feature (server feature, client feature)	Y	Y

Y: Can be installed

1.2 Installation Types

The following installation types are available for Fujitsu Enterprise Postgres:

- New installation
- Reinstallation
- Multi-version installation

1.2.1 New Installation

In initial installation, Fujitsu Enterprise Postgres is installed for the first time.

1.2.2 Reinstallation

Perform reinstallation to repair installed program files that have become unusable for any reason.

1.2.3 Multi-Version Installation

Fujitsu Enterprise Postgres products can be installed on the same server if the product version (indicated by "*x*" in "*x* SP*z*") is different from that of any version of the product that is already installed.

1.3 Installation Procedure

The following installation procedures are available for Fujitsu Enterprise Postgres:

- Installation in interactive mode
- Installation in silent mode

Select the installation procedure that corresponds to your environment.

Note

If you have antivirus software installed, the server may crash, fail to start, or stop responding, during installation or when starting up after installation. Set scan exception settings for the installation directory and resource allocation directory so that the files in these directories are not scanned for viruses.

1.3.1 Installation in Interactive Mode

Interactive mode enables installation to be performed while the required information is entered interactively.

In the interactive mode installation, the installation state of Fujitsu Enterprise Postgres is determined automatically. Install Fujitsu Enterprise Postgres using one of the following installation types in accordance with the installation state:

- New installation
- Reinstallation
- Multi-version installation

1.3.2 Installation in Silent Mode

Silent mode enables installation to be performed without the need to enter any information interactively.

New installations and multi-version installations can be performed in silent mode.

1.4 Uninstallation

Uninstallation removes the system files of the installed Fujitsu Enterprise Postgres.

Chapter 2 Operating Environment

This chapter describes the operating environment required to use Fujitsu Enterprise Postgres.

See

Refer to "Operating Environment" in the Installation and Setup Guide for Client when installing the Fujitsu Enterprise Postgres client feature at the same time.

2.1 Required Operating System

One of the operating systems shown below is required in order to use Fujitsu Enterprise Postgres.

Table 2.1 Operating systems

Operating system name
- Microsoft(R) Windows Server(R) 2016 Datacenter - Microsoft(R) Windows Server(R) 2016 Standard - Microsoft(R) Windows Server(R) 2016 Essentials
- Microsoft(R) Windows Server(R) 2019 Datacenter - Microsoft(R) Windows Server(R) 2019 Standard - Microsoft(R) Windows Server(R) 2019 Essentials
- Microsoft(R) Windows Server(R) 2022 Datacenter - Microsoft(R) Windows Server(R) 2022 Standard - Microsoft(R) Windows Server(R) 2022 Essentials

Note

- The following components of Windows Server(R) 2016, Windows Server(R) 2019 and Windows Server(R) 2022 are not supported:
  - Server Core
  - Nano Server
  - Windows Server Container
- The TCP/IP protocol must be installed.

2.2 Related Software

The following table lists the software required to use Fujitsu Enterprise Postgres.

Table 2.2 Related software

No.	Software name	Version	Product name of Fujitsu Enterprise Postgres		Remarks
			AE	SE	
1	Visual Studio	2015 2017 2019 2022	Y	Y	Required when using applications that are developed in Visual Studio.

No.	Software name	Version	Product name of Fujitsu Enterprise Postgres		Remarks
			AE	SE	
2	.NET Framework	.NET 6.0 .NET 5.0 .NET Core 3.1 .NET Standard 2.0 .NET Standard 2.1	Y	Y	
3	Perl	5.26	Y	Y	Required when using PL/Perl.
4	Python	3.9.x	Y	Y	Required when using PL/Python based on Python 3.
5	Tcl	8.6	Y	Y	Required when using PL/Tcl.

Y: Can be used

Note

- The following programs are installed during installation of Fujitsu Enterprise Postgres:
  - Microsoft Visual C++ 2015-2019 Redistributable version 14.24.28127.4

  Do not uninstall the above programs as they are required for running Fujitsu Enterprise Postgres.

The following table lists client that can be connected to the Fujitsu Enterprise Postgres server feature.

Table 2.3 Connectable client

OS	Product name
Windows Linux	Fujitsu Enterprise Postgres Client 15 or later

Note

The connection from a client product of a different version to this server function depends on the compatibility of each function included in the client product with PostgreSQL, so some functions may not be available.

The following table lists server assistant that can be connected to the Fujitsu Enterprise Postgres server feature.

Table 2.4 Connectable server assistant

OS	Product name
Windows Linux	- Fujitsu Enterprise Postgres Advanced Edition 15 - Fujitsu Enterprise Postgres Standard Edition 15

2.3 Excluded Software

This section describes excluded software.

Fujitsu Enterprise Postgres

Fujitsu Enterprise Postgres cannot be installed if all the following conditions are met:

- The product version (indicated by "*x*" in "*x* SP*z*") of the product to be installed is the same as that of the installed product

- The editions are different

Example

In the following cases, Fujitsu Enterprise Postgres cannot be installed as an exclusive product:

- The installed product is Fujitsu Enterprise Postgres Standard Edition (64bit) 15
- The product to be installed is Fujitsu Enterprise Postgres Advanced Edition (64bit) 15

Other products

There are no exclusive products.

2.4 Required Patches

There are no required patches.

2.5 Hardware Environment

The following hardware is required to use Fujitsu Enterprise Postgres.

Memory

256 MB or more is recommended (at least 128 MB is required).

2.6 Disk Space Required for Installation

The following table shows the disk space requirements for new installation of Fujitsu Enterprise Postgres. If necessary, increase the size of the file system.

Table 2.5 Disk space required for installation

Directory	Required disk space (Unit: MB)
Windows system drive	7 + 266 (*1) + 18 (*2)
Installation destination of the Server	834
Installation destination of the WebAdmin	741
Installation destination of the Client (32bit)	180
Installation destination of the Client (64 bit)	980

*1: Required for the installation of the Uninstall (middleware) tool.

*2: Required for the installation of FJQSS.

2.7 Supported System Environment

This section describes the supported system environment.

2.7.1 TCP/IP Protocol

Fujitsu Enterprise Postgres supports version 4 and 6 (IPv4 and IPv6) of TCP/IP protocols.

Note

Do not use link-local addresses if TCP/IP protocol version 6 addresses are used.

2.7.2 File System

You can install Fujitsu Enterprise Postgres only if the system folder is an NTFS volume.

2.8 PostgreSQL Version Used for Fujitsu Enterprise Postgres

Fujitsu Enterprise Postgres is based on PostgreSQL 15.0.

2.9 Notes on Using Streaming Replication

To use streaming replication, build the primary server and all standby servers using the same Fujitsu Enterprise Postgres version (*1).

*1: The product version is indicated by "*x*" in the notation "*x* SP*z*".

Note

Streaming replication cannot be used in combination with Open Source PostgreSQL.

Chapter 3 Installation

This chapter describes the procedures for the installation of Fujitsu Enterprise Postgres.

Note

- The installation must be performed by a user with administrator privileges (a user ID that belongs to the Administrators group).
- Stop all applications before starting the installation.
- The Windows Installer service must be running.
- The remote desktop service is installed in application server mode, it is necessary to switch to install mode by executing the command shown below before installation. Also, after the installation is completed, execute the command shown below to switch back to execute mode.

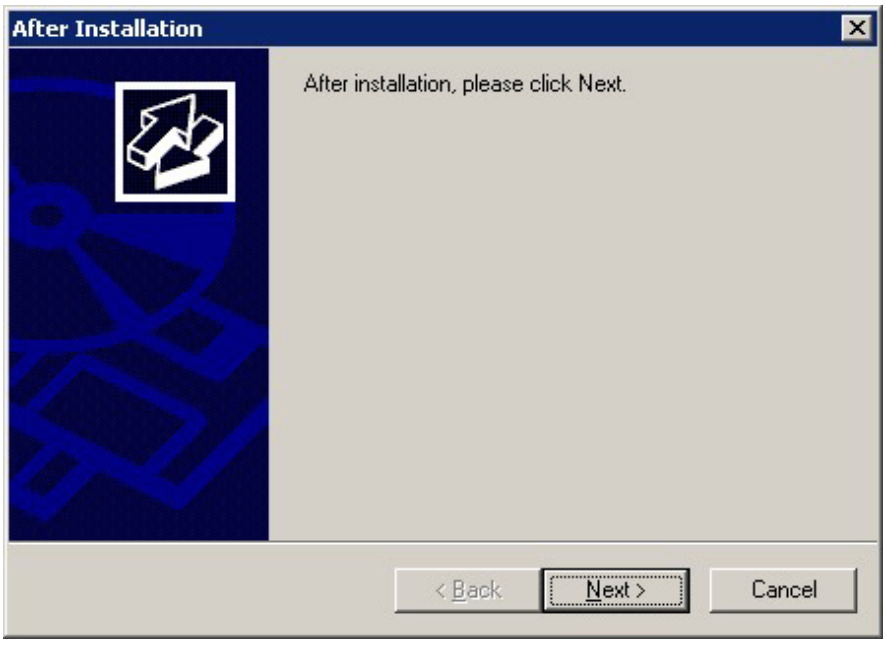
  [Before the installation]

```
CHANGE USER /INSTALL
```

  [After the installation]

```
CHANGE USER /EXECUTE
```

- When installing the Fujitsu Enterprise Postgres Client (32 bit), do not specify a destination folder under the environment ProgramFiles variable.
- The following window may be displayed when executing the installation program:



If this window is displayed, perform the following operations:

1. Perform the installation steps until the [InstallShield Wizard Complete] window is displayed.
2. At the window shown above, click [Next].

3. The window shown below is displayed. Click [Finish].

Information

- If a [User Account Control] dialog box is displayed at the start of the installation, click [Yes] to continue processing:

  If [No] is clicked, permission to continue is denied and an [Error] dialog box will be displayed.
  To continue the installation, click [Retry] at the [Error] dialog box. To end the installation, click [Cancel].

- If installation is suspended or processing terminates abnormally, the [Program Compatibility Assistant] dialog box may be displayed. Click [This program installed correctly] and continue operation.

3.1 Pre-installation Tasks

Check the system environment below before installing Fujitsu Enterprise Postgres.

Check the disk space

Ensure that there is sufficient disk space to install Fujitsu Enterprise Postgres.

Refer to "2.6 Disk Space Required for Installation" for information on the required disk space.

Check the installed products and determine the installation method

In Windows, click [All Programs] or [All apps], then [Fujitsu], and then [Uninstall (middleware)]. In the displayed window, check the installed products.

If Fujitsu Enterprise Postgres is already installed, determine which installation method to use:

- Reinstallation
- Multi-version installation

Remove applied updates

If you perform reinstallation as the installation method, remove applied updates using the procedure shown below.

Note

If a product is installed without removing applied updates, the following problems will occur:

- Performing reinstallation

  If an update with the same update and version number is applied, an error informing you that the update has already been applied is displayed.

1. Display the applied updates

   Execute the following command to display the applied updates:

```
C:\>uam showup
```

2. Remove the updates

   Execute the command below to remove the updates. If an update with the same update number was applied more than once, the updates are removed in order, starting from the highest version number.

```
C:\>uam remove -i update-number
```

Determine the preferred WebAdmin configuration

Starting with Fujitsu Enterprise Postgres 9.5, WebAdmin can be installed in two configurations:

- Single-server
- Multiserver

See

Refer to "Appendix H Determining the Preferred WebAdmin Configuration" for details.

3.2 Installation in Interactive Mode

The installation must be performed by a user with administrator privileges (a user ID that belongs to the Administrators group).

Note

When reinstalling the product, back up the following folder in which the WebAdmin instance management information is stored:

webAdminInstallFolder\data\fepwa

Follow the procedure below to perform the backup.

1. Stop the WebAdmin server. Refer to "B.1.3 Stopping the Web Server Feature of WebAdmin" for details.
2. Back up the following folder:

   webAdminInstallFolder\data\fepwa

Replace the above folder with the backed up folder when the reinstallation is complete.

Point

For installation in interactive mode, default values are set for the installation information. The following settings can be changed for a new installation or a multi-version installation:

- Installation destination
  - It is necessary to specify a local disk as the installation destination of Fujitsu Enterprise Postgres.
  - If using WebAdmin, do not use fullwidth characters or halfwidth katakana characters in [Installation destination folder].
- WebAdmin setup information, if WebAdmin is selected
  To change the port number, confirm that it is an unused port number between 1024 and 49151. Additionally, take note of the Web server port number for the Windows Firewall settings.

The installation procedure is described below.

1. Stop applications and programs

When reinstalling the product, all applications and programs that use the product must be stopped.

Before starting the installation, stop the following:

- Applications that use the product
- Connection Manager
- Instance
- Web server feature of WebAdmin

  If you are using WebAdmin, stop WebAdmin.

  Refer to "B.1.3 Stopping the Web Server Feature of WebAdmin" for details.
- Mirroring Controller

  Execute the mc_ctl command with the stop mode option specified and stop the Mirroring Controller.

  Example

  ```
  > mc_ctl stop -M D:\mcdir\inst1
  ```
- pgAdmin

2. Inserting the DVD

Insert the Fujitsu Enterprise Postgres DVD into the drive.

3. Run the installation

The installation menu will be displayed. Click [Installation].

If the Autorun feature of Windows is disabled, or a remote desktop service (terminal service) is used, the installation program is not automatically started. Execute the following file using [Run] or Windows Explorer.

```
Z:\autorun.exe
```

Z: The drive into which the DVD is inserted.

4. Select the products to install

The [Installation product] window will be displayed.

Select the products to install, and then click [Next].

If a selected product can only be reinstalled, refer to "6. Check the settings".

Information

- To develop or execute a 32-bit application in a 64-bit environment, Fujitsu Enterprise Postgres Client (32bit) is required.
- The Fujitsu Enterprise Postgres server component and WebAdmin can be installed on the same machine by selecting the "Fujitsu Enterprise Postgres server component" and the "WebAdmin component".
- If the selected product has been installed, the [Select installation method] window is displayed for each product. To perform a multi-version installation, click [Next].

5. Checking the installation content

The [Confirm installation] window will be displayed.

Click [Next] to start the installation.

To modify the settings, select [Modify], and then click [Next]. Follow the on-screen instructions.

If you have not set up WebAdmin during installation, refer to "Appendix B Setting Up and Removing WebAdmin for details.

Note

If using WebAdmin for operation, make a note of the Web server port number displayed in the settings, for use in the Windows firewall settings.

6. Check the settings

The [Confirm installation] window is displayed for reinstallation, or if the installation information is modified.

Click [Install] to start the installation.

To change any settings, click [Back].

7. Completing installation

The [Installation complete] window is displayed. Click [Finish].

From [All Programs] or [All apps], click [Fujitsu] >> [Uninstall (middleware)]. If the installed product names have been added under [Software Name], installation is complete.

3.3 Installation in Silent Mode

Installation in silent mode can be performed only when the installation method is one of the following:

- New installation
- Multi-version installation

See

Refer to the Fujitsu Enterprise Postgres product website for information on installation in silent mode, such as the installation parameters and error messages.

The installation procedure is described below.

1. Insert the DVD

Insert the "server program" DVD in the DVD drive.

The [Install Menu] window will be displayed automatically. Click [Finish].

2. Create an installation parameters CSV file

Consider the features that will be required for system operations, and then create an installation parameters CSV file that uses the following specification format.

```
sectionName, parameterName, value
sectionName, parameterName, value
    :
```

Note

If using WebAdmin for operation, make a note of the Web server port number displayed in the settings (the port number defined in WebPortNumber1), for use in the Windows firewall settings.

Information

The template for the installation parameters CSV file is "*Z*:\sample\sample.csv" (*Z* is the drive into which the DVD is insered).

3. Start the command prompt

In Windows, right-click [Command Prompt] and then select [Run as administrator].

4. Run the installation

Execute the command below.

```
Z:\>silent.bat c:\temp\inspara.csv
```

Z: The drive into which the DVD is inserted.

Also in the example above, c:\temp\inspara.csv is the installation parameter CSV file name.

If the installer ends in an error, a message is output to the log file and return values are returned.

Chapter 4 Setup

This chapter describes the setup procedures to be performed after installation completes.

4.1 Operating Method Types and Selection

This section describes how to operate Fujitsu Enterprise Postgres.

There are two methods of managing Fujitsu Enterprise Postgres operations - select one that suits your purposes:

The Operation Guide describes the operating method using WebAdmin, and the equivalent operating method using the server commands.

Simple operation management using a web-based GUI tool (WebAdmin)

Suitable when using frequently used basic settings and operations for operation management.

This method allows you to perform simple daily tasks such as starting the system before beginning business, and stopping the system when business is over, using an intuitive operation.

Usage method

Usage is started by using WebAdmin to create the instance.

By using an external scheduler and the pgx_dmpall command, periodic backups can be performed, which can then be used in recovery using WebAdmin.

Note

Do not use a server command other than pgx_dmpall and pgx_keystore or a server application. Operation modes that use server commands and server applications cannot be used in conjunction with WebAdmin. If used, WebAdmin will not be able to manage the instances correctly.
In addition, to perform a backup by copy command from the pgx_dmpall command, select the operating method using the server commands.
Refer to Reference and the PostgreSQL Documentation for information on server commands and server applications.

Advanced operation management using server commands

When operating in a system that is automated by operation management middleware (Systemwalker Centric Manager, for example), this method allows you to use more detailed settings and operations and perform higher level operation management.

An overview of the operating method using the GUI, and its relationship with the operating method using the server commands, are shown below.

Refer to the Operation Guide for details.

Operation		Operation with the GUI	Operation with commands
Setup	Creating an instance	WebAdmin is used. The server machine capacity, and the optimum parameter for operations using WebAdmin, are set automatically.	The configuration file is edited directly using the initdb command.
	Creating a standby instance	WebAdmin is used. WebAdmin performs a base backup of the source instance and creates a standby instance.	A standby instance is created using the pg_basebackup command.
	Changing the configuration files	WebAdmin is used.	The configuration file is edited directly.

Operation		Operation with the GUI	Operation with commands
Starting and stopping an instance		WebAdmin is used.	The net command or sc command of the operating system is used.
Creating a database		This is defined using pgAdmin of the GUI tool, or using the psql command or the application after specifying the DDL statement.	
Backing up the database		WebAdmin, or the pgx_dmpall command, is used.	It is recommended that the pgx_dmpall command be used. Recovery to the latest database can be performed.
Database recovery		WebAdmin is used.	To use the backup that was performed using the pgx_dmpall command, the pgx_rcvall command is used.
Monitoring	Database errors	The status in the WebAdmin window can be checked. (*1)	The messages that are output to the database server log are monitored (*1)
	Disk space	The status in the WebAdmin window can be checked. A warning will be displayed if the free space falls below 20%. (*1)	This is monitored using the fsutil command (check free space), and the dir command (check used space), of the operating system, for example. (*1)
	Connection status	This can be checked using pgAdmin of the GUI tool, or referencing pg_stat_activity of the standard statistics view from psql or the application.	

*1: This can be used together with system log monitoring using operations management middleware (Systemwalker Centric Manager, for example).

Refer to "Periodic Operations" and "Actions when an Error Occurs" in the Operation Guide for information on monitoring and database recovery.

4.2 Preparations for Setup

This section describes the preparation required before setting up Fujitsu Enterprise Postgres.

4.2.1 Creating an Instance Administrator

Decide the OS user account that will become the instance administrator. Use either a new user, or a user that already exists.

To create a user in Windows, select [Administrative Tools], [Computer Management], and then create the user in [Local Users and Groups]. Refer to "Help and Support" for details.

The following characters can be used for user names:

- - (hyphen)
- _ (underscore)
- Space
- A-Z, a-z, 0-9 (alphanumeric)

Note

The following notes apply if using WebAdmin for operations:

- The instance administrator must have a local OS user account.
- After creating the user account of the instance administrator, log in to the operating system. A profile directory is created for the user when logging in to the operating system for the first time. This directory will be used by WebAdmin.
- If changing the password for the user account of the instance administrator, always ensure to stop the instance and log out of WebAdmin before making the change. If you mistakenly change the password while logged in to WebAdmin or while the instance is running, log out from WebAdmin, and then log in again, and stop and start the instance.
- If the password is changed for the user account of the instance administrator, set the changed password using ALTER ROLE WITH ENCRYPTED PASSWORD.

4.2.1.1 Security policy settings

If using commands for operation, security settings that allow logon as a service are required for the operating system user account of the instance administrator in order to start and stop an instance using a Windows service.

Information

If using WebAdmin for operation, these settings are not required as WebAdmin performs the settings automatically for the user ID (operating system user account) that logged in to the database server.

The following explains how to perform the security settings to allow logon as a service:

1. Displaying the Local Security Policy window

   In Windows, select [Administrative Tools], and then click [Local Security Policy].

2. Setting up security

   1. In the [Local Security Policy] window, select [Security Settings], select [Local Policies], and then click [User Rights Assignment].
   2. Under [Policy] in the [User Rights Assignment] window, double-click [Log on as a service].
   3. In the [Log on as a service Properties] window, set the following:
      a. Select the [Local Security Setting] tab.
      b. On the [Local Security Setting] tab, click [Add User or Group].
      c. In the [Select Users or Groups] window, enter the operating system user account of the instance administrator in [Enter the object names to select].
      d. Click [OK].
   4. In the [Log on as a service Properties] window, click [OK].

4.2.2 Preparing Directories for Resource Deployment

Prepare the directories required when creating instances.

Considerations when deploying resources

The disk configuration on the resource deployment destination is important, because it affects not only recovery following disk corruption, but normal operation as well. The points for determining the disk configuration are as follows:

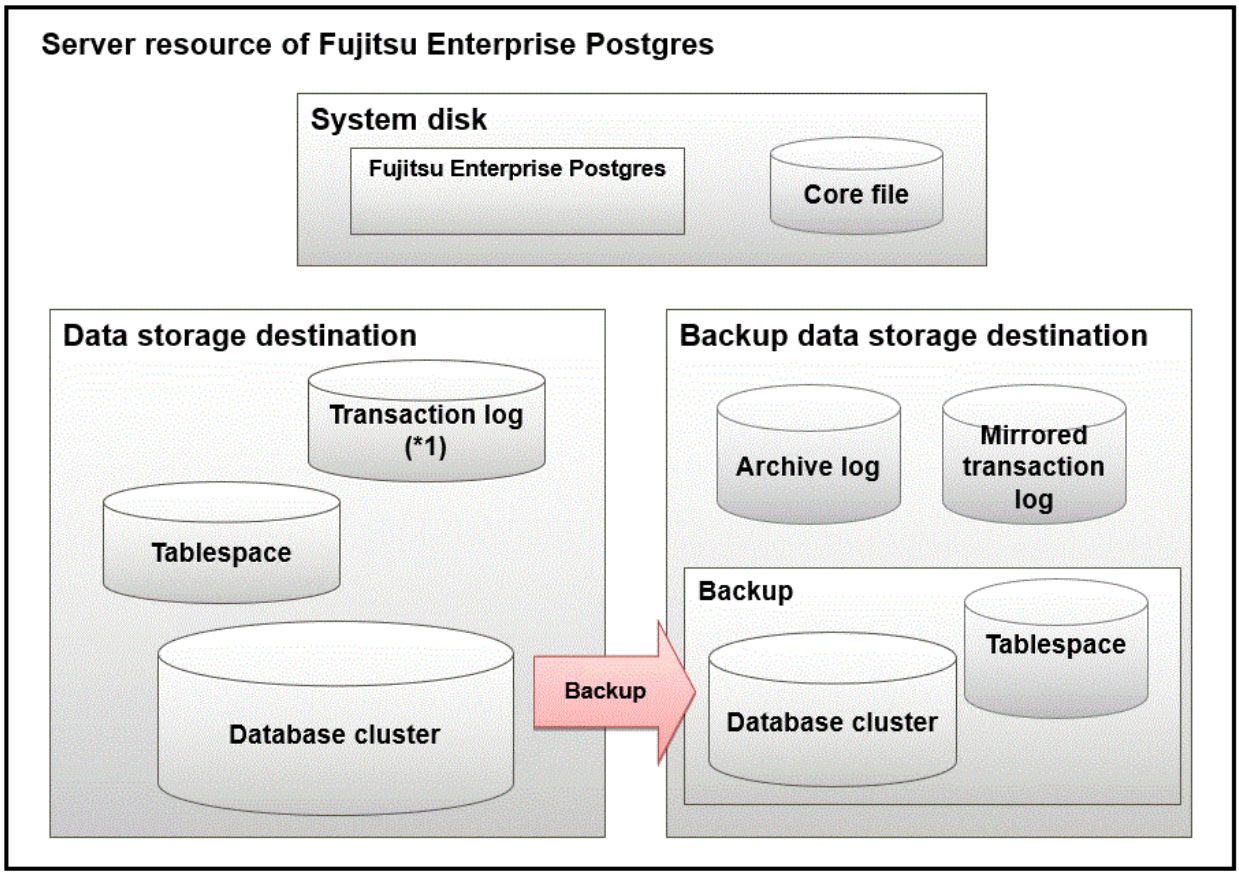
1. If the backup data storage destination and the data storage destination are both lost, it will not be possible to recover the data, so deploy them to separate disks.
2. To shorten the recovery time following a single disk fault, deploy the system disk and data storage destination to separate disks.

3. The backup data storage destination requires at least double the capacity of the data storage destination, so deploy it to the disk with the most space available.

4. When large amounts of data are updated, the write-to load for the data storage destination, transaction log storage destination, and backup data storage destination (mirrored transaction log) will also be great. For this reason, deploy them to separate disks, out of consideration for performance.

Note

- When using the volume manager provided by the operating system, be aware of which physical disk the file system has been created on, for example, by deploying the data storage destination and the backup data storage destination to separate disks.
- If using WebAdmin, specify an NTFS volume for the data storage destination and backup data storage destination. A network drive cannot be specified.



*1: To distribute the I/O load, place the transaction log on a different disk from the data storage destination.

Resource	Role
Database cluster	The area where the database is stored. It is a collection of databases managed by an instance.
Tablespace	Stores table files and index files in a separate area from the database cluster. Specify a space other than that under the database cluster.
Transaction log	Stores log information in preparation for a crash recovery or rollback. This is the same as the WAL (Write Ahead Log).
Archive log	Stores log information for recovery

Resource	Role
Mirrored transaction log (mirrored WAL)	Enables a database cluster to be restored to the state immediately before an error even if both the database cluster and transaction log fail when performing backup/recovery operations using the pgx_dmpall command or WebAdmin.
Corefile	Fujitsu Enterprise Postgres process corefile output when an error occurs with a Fujitsu Enterprise Postgres process.

Examples of disk deployment

The following are examples of disk deployment:

Number of disks	Disk	Deployment
3	System disk	Fujitsu Enterprise Postgres program
		Corefile
	Connected physical disk	Data storage destination, transaction log storage destination
	Connected physical disk	Backup data storage destination
2	System disk	Fujitsu Enterprise Postgres program
		Corefile
		Data storage destination, transaction log storage destination
	Connected physical disk	Backup data storage destination

Proposal for disk deployment using WebAdmin

To generate an instance using WebAdmin, we recommend an optimum deployment that takes into account the status of all disks at the time of instance generation, and items 1 to 3 in the "Considerations when deploying resources" subheading above, based on the limitations below (note that a different deployment can also be specified).

- The instance administrator has read and write privileges for the volumes.

Preparing directories

The directories to be prepared depend on the way that you create the instances.

The following shows the directories that need to be prepared:

Directory to be prepared	Using WebAdmin	Using the initdb command
Data storage destination	Y (*1)	Y
Backup data storage destination	O (*1) (*4)	O
Transaction log storage destination	O (*1) (*2)	O
Corefile output destination	N (*3)	O

Y: Required
O: Optional
N: Not required

*1: WebAdmin automatically creates a directory

*2: The default is to create in a directory in the data storage destination. When it is necessary to distribute the I/O load for the database data and the transaction log, consider putting the transaction log storage destination on a different disk from the data storage destination

*3: The corefile path is as follows:

*userProfileFolder**localSettingsFolder*\Fujitsu\fsep_*version**instanceNamePortNumber*\core

version: product version_WA_architecture

Note: The product version is normally the version of WebAdmin used to create the instance. For example, WebAdmin 15 allows a user to create a FUJITSU Enterprise Postgres 9.6 instance on a database server having WebAdmin 9.6. In this case, because WebAdmin 9.6 is used to create the instance, the product version will be "96".

PortNumber: port number specified when creating the instance

Example

C:\Users\naomi\AppData\Local\Fujitsu\fsep_150_WA_64\myinst27599\core

To change the output destination, specify in the core_directory parameter and core_contents parameter in postgresql.conf. Refer to "Parameters" in the Operation Guide for information on the settings for these parameters.

*4: This directory is required when instance backup is enabled.

- The directories must meet the following conditions:
  - The directory owner must be the OS user account that you want to be the instance administrator
  - The directory must have write permission
  - The directory must be empty
- If using WebAdmin, network drives cannot be used.
- If using WebAdmin, the following halfwidth characters can be used for directory names:
  - \(backslash)
  - - (hyphen)
  - _ (underscore)
  - : (colon)
  - Space
  - A-Z, a-z, 0-9 (alphanumeric)
- If anti-virus software is used, set scan exception settings for folders so that none of the server resources that comprise Fujitsu Enterprise Postgres are scanned for viruses. Alternatively, if the server resources that comprise Fujitsu Enterprise Postgres are to be scanned for viruses, stop the instance and perform the scan when tasks that use Fujitsu Enterprise Postgres are not operating.

Confirm and configure directory access permissions

If the instance administrator user has "Administrator" permissions (user ID belonging to the Administrators group), it is necessary to configure the settings so that each directory inherits the file and directory access permissions for the instance administrator user.

Therefore, ensure that the setting to inherit permissions has been configured.

The following is an explanation on how to confirm and configure the settings.

How to confirm access permissions

Perform the following operations in Windows Explorer on the directories to be prepared in advance:

1. Right-click on the applicable directory, and then click [Properties] from the menu that is displayed.
2. In the [*applicableDir* Properties] window, select [Security] >> [Advanced].
3. In the [Advanced Security Settings for *applicableDir*] window, and in the [Permission entries] list under the [Permissions] tab, confirm that [Applies to] for the instance administrator user is "This folder, subfolders and files".
4. Click [OK].

How to configure the access permissions

Perform the following operations in Windows Explorer if there are any directories that have not been configured for the access permissions to be inherited.

1. Right-click on the applicable directory, and then click [Properties] from the menu that is displayed.
2. In the [*applicableDir* Properties] window, select [Security] >> [Advanced].
3. In the [Advanced Security Settings for *applicableDir*] window, click [Add].
4. In the [Permission Entry for *applicableDir*] window, click [Select a principal].
5. In the [Select User or Group] window, enter the instance administrator user name as the object name to select, and then click [OK].
6. In the [Permission Entry for *applicableDir*] window, set [This folder, subfolders and files] for [Applies to:], and under [Basic permissions], allow read and write permissions, and then click [OK].
7. In the [Advanced Security Settings for *applicableDir*] window, confirm that the instance administrator user has been added, with [This folder, subfolders and files] set for [Applies to] in the [Permission entries] list.
8. Click [OK].

Information

The access permissions can also be configured using the icacls command provided by the operating system.

The following is an execution example in which the application destination is set to "(OI)(CI)" and the access permissions are set to "(F) (Full access permissions)" when the data storage destination is "D:\database\inst1" and the instance administrator user is "fsepuser":

```
>icacls D:\database\inst1 /grant fsepuser:(OI)(CI)(F)
processed file: D:\database\inst1
Successfully processed 1 files; Failed processing 0 files
```

4.2.3 Estimating Resources

Estimate the resources to be used on the Fujitsu Enterprise Postgres.

Refer to "Appendix E Estimating Database Disk Space Requirements" for information on estimating database disk space requirements.

Refer to "Parameters automatically set by WebAdmin according to the amount of memory" when creating multiple instances with WebAdmin.

Refer to "Appendix F Estimating Memory Requirements" when creating instances with the initdb command, to estimate memory usage.

4.2.4 Windows Firewall Settings

This section explains the Windows firewall settings required if using WebAdmin for operation.

These settings are not required if using server commands for operation.

If the Windows firewall feature is to be enabled, you should enable a port number on the Web server. The following explains how to enable a port number:

Windows Server(R) 2016:

1. Select [Systems and Security] from [Control Panel] and click [Windows Firewall].
2. In the [Windows Firewall] window, click [Advanced settings].
3. In the [Windows Firewall with Advanced Security] window, click [Inbound Rules] on the left side of the window.
4. Click [New Rule] on the right side of the window.
5. In the [New Inbound Rule Wizard] window, select [Port], and then click [Next].

6. Select [TCP] and [Specific local ports], then specify the Web server port number specified during the WebAdmin setup, and then click [Next].
7. Select [Allow the connection], and then click [Next].
8. Select the profiles for which this rule applies, and then click [Next].
9. In [Name], specify the desired name, and then click [Finish].
10. In the [Windows Firewall with Advanced Security] window, check if the added rule is enabled under [Inbound Rules] in the center of the window.

In cases other than the above:

1. Select [Systems and Security] from [Control Panel] and click [Windows Defender Firewall].
2. In the [Windows Defender Firewall] window, click [Advanced settings].
3. In the [Windows Defender Firewall with Advanced Security] window, click [Inbound Rules] on the left side of the window.
4. Click [New Rule] on the right side of the window.
5. In the [New Inbound Rule Wizard] window, select [Port], and then click [Next].
6. Select [TCP] and [Specific local ports], then specify the Web server port number specified during the WebAdmin setup, and then click [Next].
7. Select [Allow the connection], and then click [Next].
8. Select the profiles for which this rule applies, and then click [Next].
9. In [Name], specify the desired name, and then click [Finish].
10. In the [Windows Defender Firewall with Advanced Security] window, check if the added rule is enabled under [Inbound Rules] in the center of the window.

4.2.5 Preparing for Output to the Event Log

This section provides an explanation on the preparation to be carried out if you are outputting error logs to the event log.

If outputting error logs to the event log, you should register an event source name beforehand.

If you do not register an event source name, the message content output to the event log may be incomplete.

Due to the default event source name "FUJITSU Enterprise Postgres Server" being output to the event log when using the following commands, you should register this default event source name beforehand:

- pg_ctl command
- pgx_dmpall command
- pgx_rcvall command

The following is an example in which the DLL of a 64-bit product is registered under the default event source name:

```
> regsvr32  "C:\Program Files\Fujitsu\fsepv<x>server64\lib\pgevent.dll"
```

Note that this step is not required if using WebAdmin to create an instance.

If using multiple instances

You can output messages corresponding to the event source name assigned by the user, so that messages output to the event log can be identified by instance.

The following is an example in which the DLL of a 64-bit product is registered under the event source name "Fujitsu Enterprise Postgres inst1":

```
> regsvr32 /n /i:"Fujitsu Enterprise Postgres inst1" "C:\Program Files\Fujitsu\fsepv<x>server64\lib
\pgevent.dll"
```

You will need to edit the parameters for each instance, therefore, after creating an instance, refer to "4.5.1 Error Log Settings" when performing this setting.

If installing multiple versions

If Fujitsu Enterprise Postgres is already installed on the same machine, search for the key below in Registry Editor, and make a note of the path of the registered DLL. Afterwards, register a new DLL under the default event source name.

Use the DLL path that you made a note of in the above step when re-registering the default event source name during an uninstall.

FUJITSU Enterprise Postgres Server

See

Refer to "Registering Event Log on Windows" in "Server Setup and Operation" in the PostgreSQL Documentation for information on how to register event source names.

4.3 Creating Instances

There are two methods that can be used to create an instance:

- 4.3.1 Using WebAdmin
- 4.3.2 Using the initdb Command

Creating multiple instances

Multiple instances can be created.

The memory allocated needs to be adjusted when multiple instances are created with WebAdmin (refer to "Parameters automatically set by WebAdmin according to the amount of memory" for details).

Features that cannot be set up using WebAdmin

The "Storage data protection using transparent data encryption" feature cannot be set up using WebAdmin.

To set up this feature in an instance created by WebAdmin, perform the additional setup tasks detailed in "Storage Data Protection using Transparent Data Encryption" in the Operation Guide.

Note

- Instances created using the initdb command (command line instances) can be managed using WebAdmin, however, they must first be imported into WebAdmin. Refer to "4.3.1.4 Importing Instances" for details.
- Always use WebAdmin to delete instances that were created or imported using WebAdmin. Because WebAdmin management information cannot be deleted, WebAdmin will determine that the instance is abnormal.
- Databases with the names 'template0' and 'template1' are automatically created when an instance is created.
  These databases are used as the templates for databases created later. Furthermore, a default database with the name 'postgres' is automatically created, which will be used with Fujitsu Enterprise Postgres commands. It is important that you do not delete these databases created by default.
- When an instance that uses WebAdmin is created successfully, the following Windows service is registered:

  `fsep_version_WA_architecture_userName_instanceNamePortNumber`

  The account and password of the instance administrator are registered in the Windows service.
  If the password for this account is changed, you must also change the password registered in the service.
  Change this at the Properties window registered in the Windows service.

  Note: The product version is normally the version of WebAdmin used to create the instance. In addition, WebAdmin 15 also allows a user to create a Fujitsu Enterprise Postgres 9.6 instance on a database server having WebAdmin 9.6. In this case the product version will be "96".

- Refer to "4.5.2 Configuring Automatic Start and Stop of an Instance" for information on how to start and stop the operating system of the database server, and how to start and stop linked instances.

4.3.1 Using WebAdmin

This section describes how to create an instance using WebAdmin.

WebAdmin must be set up correctly before it can be used. Refer to "B.1 Setting Up WebAdmin" for details. Additionally, if WebAdmin needs to be configured to use an external repository database, refer to "B.3 Using an External Repository for WebAdmin" for details.

It is recommended to use the following browsers with WebAdmin:

- Microsoft Edge (Build41 or later)

WebAdmin will work with other browsers, such as Firefox and Chrome, however, the look and feel may be slightly different.

Configure your browser to allow cookies and pop-up requests from the server on which Fujitsu Enterprise Postgres is installed.

Refer to "Appendix A Recommended WebAdmin Environments" for information on how to change the pop-up request settings and other recommended settings.

Note

- WebAdmin does not run in Windows(R) safe mode.
- If the same instance is operated from multiple WebAdmin windows, it will not work correctly.
- If the same instance is operated from multiple WebAdmin versions, it will not work correctly. Always use the latest version of WebAdmin for instance operations.
- For efficient use of WebAdmin, it is recommended not to use the browser [Back] and [Forward] navigation buttons, the [Refresh] button, and context-sensitive menus, including equivalent keyboard shortcuts.
- Copying and pasting the WebAdmin URLs are not supported. Additionally, bookmarking of WebAdmin URLs is not supported.
- It is recommended to match the language between the instance server locale and WebAdmin.
- WebAdmin supports only two languages: English and Japanese.
- It is recommended to change the WebAdmin language setting from the instance details page only.
- It is recommended to operate WebAdmin using the WebAdmin launcher.
- WebAdmin uses the labels "Data storage path", "Backup storage path" and "Transaction log path" to indicate "data storage destination", "backup data storage destination" and "transaction log storage destination" respectively. In this manual these terms are used interchangeably.
- If the browser was not operated for a fixed period (about 30 minutes), the session will time out and the login page will be displayed again for the next operation.
- Port access permissions
  If a port is blocked (access permissions have not been granted) by a firewall, enable use of the port by granting access. Refer to the vendor document for information on how to grant port access permissions.
  Consider the security risks carefully when opening ports.
- When creating or importing an instance in WebAdmin, set the log_directory parameter in postgresql.conf in the following format:
  log_directory='*userProfileFolder**localSettingsFolder*\\Fujitsu\\fsep_*version**instanceNamePortNumber*\\log'

  Example: *userProfileFolder**localSettingsFolder* will be C:\\Users*userName*\\AppData\\Local.

4.3.1.1 Logging in to WebAdmin

This section describes how to log in to WebAdmin.

Startup URL for WebAdmin

In the browser address bar, type the startup URL of the WebAdmin window in the following format:

```
http://hostNameOrIpAddress:portNumber/
```

- *hostNameOrIpAddress*: Host name or IP address of the server where WebAdmin is installed
- *portNumber*: Port number of WebAdmin. The default port number is 27515.

Example

For a server with IP address "192.0.2.0" and port number "27515":

```
http://192.0.2.0:27515/
```

The startup screen is displayed. From this window you can log in to WebAdmin or access the product documentation.

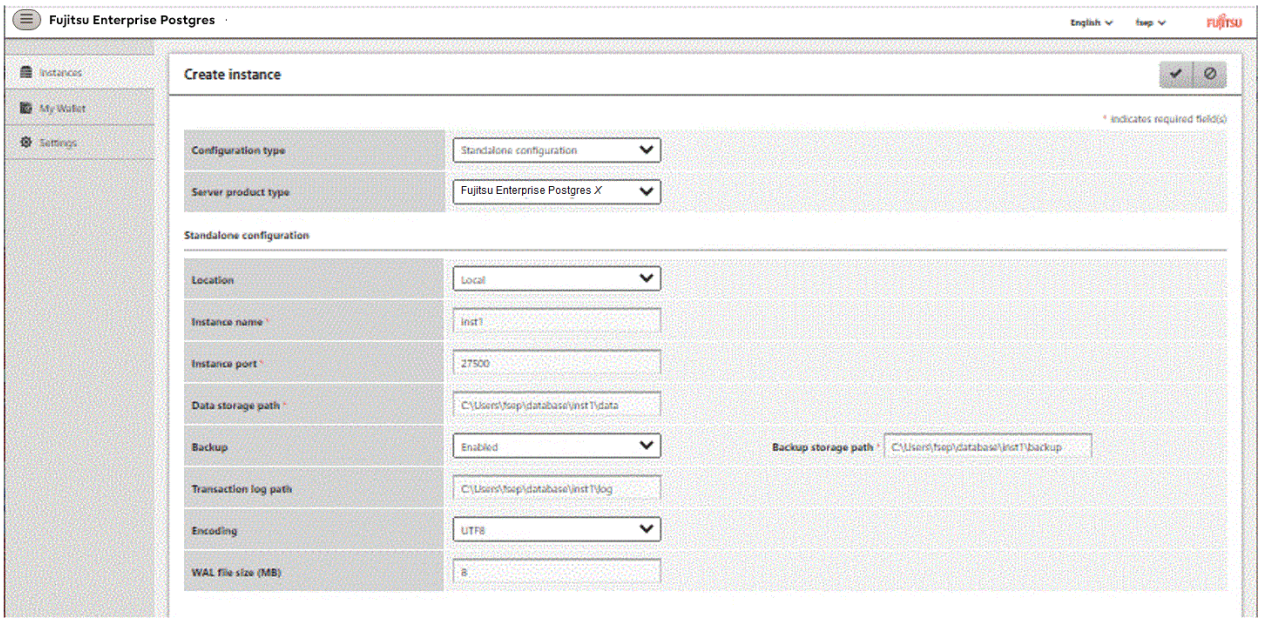
Logging in to WebAdmin

Click [Launch WebAdmin] in the startup URL window to start WebAdmin and display the login window. Enter the instance administrator user name (operating system user account name) and password, and log in to WebAdmin. User credential (instance administrator user ID and password) should not contain hazardous characters. Refer to "Appendix C WebAdmin Disallow User Inputs Containing Hazardous Characters".

4.3.1.2 Creating an Instance

This section describes how to create an instance.

1. Start WebAdmin, and log in to the database server.
2. In the [Instances] tab, click ➕.
3. Enter the information for the instance to be created.



Enter the following items:

- [Configuration type]: Whether to create a standalone instance or an instance that is part of a cluster.
- [Server product type]: Sets which of the following instances to create:
  - Fujitsu Enterprise Postgres 9.5 Instances

- Fujitsu Enterprise Postgres 9.6 Instances
- Fujitsu Enterprise Postgres 10 Instances
- Fujitsu Enterprise Postgres 11 Instances
- Fujitsu Enterprise Postgres 12 Instances
- Fujitsu Enterprise Postgres 13 Instances
- Fujitsu Enterprise Postgres 14 Instances
- Fujitsu Enterprise Postgres 15 Instances

The default is "Fujitsu Enterprise Postgres 15".
WebAdmin can create and manage instances compatible with the following, but new functionality in FUJITSU Enterprise Postgres 15 may not support the instance or it may be disabled.

- Fujitsu Enterprise Postgres 9.5
- Fujitsu Enterprise Postgres 9.6
- Fujitsu Enterprise Postgres 10
- Fujitsu Enterprise Postgres 11
- Fujitsu Enterprise Postgres 12
- Fujitsu Enterprise Postgres 13
- Fujitsu Enterprise Postgres 14

- [Location]: Whether to create the instance in the server that the current user is logged into, or in a remote server. The default is "Local", which will create the instance in the server machine where WebAdmin is currently running.
- [Instance name]: Name of the database instance to manage

  The name must meet the conditions below:

  - Maximum of 16 characters
  - The first character must be an ASCII alphabetic character
  - The other characters must be ASCII alphanumeric characters
- [Instance port]: Port number of the database server
- [Data storage path]: Directory where the database data will be stored
- [Backup]: Whether to enable or disable the WebAdmin backup feature. The default is "Enabled". Select "Disabled" to disable all backup and restore functionality for the instance. If "Enabled" is selected, enter the following item:
  - [Backup storage path]: Directory where the database backup will be stored
- [Transaction log path]: Directory where the transaction log will be stored
- [Encoding]: Database encoding system
- [WAL file size]: Allow the WAL file size to be set when creating an instance. The default is 16 MB if the field is blank. The size specified must be a power of 2 between 1 and 1024. This option is not available for standby instances.

If "Remote" is selected for [Location], enter the following additional items:

- [Host name]: Name of the host where the instance is to be created
- [Operating system credential]: Operating system user name and password for the remote machine where the instance is to be created
- [Remote WebAdmin port for standalone]: Port in which WebAdmin is accessible in the remote machine

Note

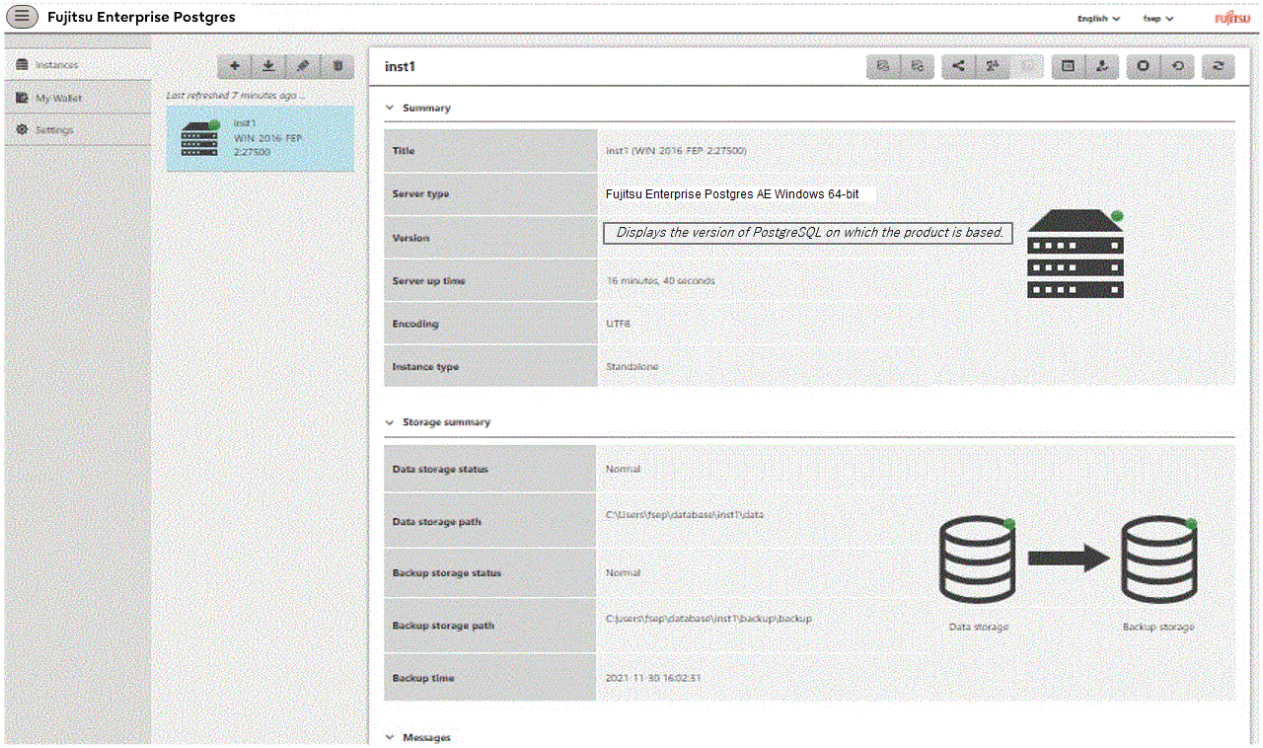
- Refer to "4.2.2 Preparing Directories for Resource Deployment" - "Considerations when deploying resources" for information on points to consider when determining the data storage path, backup storage path, and transaction log path.
- The following items can be modified after the instance has been created. These items cannot be modified on instances that have compatibility with Fujitsu Enterprise Postgres 9.5.
  - Instance name
  - Port number
  - Backup storage path

  Refer to "Editing instance information" for details.
- Do not specify shortcuts for the data storage path, backup storage path, or transaction log path.
- In the instance that is created using WebAdmin, the locale of the character set to be used in the database, and the locale of the collating sequence, are fixed using C.
- The following characters can be used for the data storage path, backup storage path, and transaction log path:
  - \ (backslash)
  - - (hyphen)
  - _ (underscore)
  - : (colon)
  - Space
  - A-Z, a-z, 0-9 (alphanumeric)
- Instance administrator read and write permissions are required for the data storage path, backup storage path, and transaction log path.
- For the port number, specify an unused port number in the following range:
  - 1024 to 49151
- Make a note of the port number for use in the Windows firewall settings.
- Refer to "4.5.2 Configuring Automatic Start and Stop of an Instance" for information on configuring the automatic start and stop of instances.
- For enhanced security, WebAdmin encrypts the superuser password using SCRAM-SHA-256 authentication for all Fujitsu Enterprise Postgres 10 or later instances. The client/driver must therefore support SCRAM-SHA-256 authentication if they need to connect to Fujitsu Enterprise Postgres 10 or later instances created by WebAdmin with superuser credentials.
- Host name and Operating system credential (Operating system user name and password) should not contain hazardous characters. Refer to "Appendix C WebAdmin Disallow User Inputs Containing Hazardous Characters".

4. Click to create the instance.

   If the instance is created successfully, a message indicating the same will be displayed.

5. The instance will be started when it is created successfully.



6. Back up the basic information that was set

   Back up the WebAdmin management information periodically to ensure operational continuity when a fault occurs on the system disk. Follow the procedure below to perform the backup.

   a. Stop the WebAdmin server. Refer to "B.1.3 Stopping the Web Server Feature of WebAdmin" for details.

   b. Back up the following directory:

      webAdminInstallDir\data\fepwa

Note

The following message is output during startup of an instance when the startup process is operating normally, therefore, the user does not need to be aware of this message.

```
FATAL:  the database system is starting up
```

4.3.1.3 Changing Instance Settings

You can change the information that is set when an instance is created.

Change the following settings to suit the operating and management environment for Fujitsu Enterprise Postgres.

- Instance configuration
  - Character encoding
  - Communication
  - SQL options
  - Memory
  - Streaming replication

- Changing client authentication information
- Editing instance information

Information

These settings are the same as the parameters that can be set in the files shown below. Refer to "Appendix D Configuring Parameters" for information on the equivalence relationship between the item name and the parameter.

- postgresql.conf
- pg_hba.conf

Note

The files shown below can also be modified directly, however if a parameter not described in "Appendix D Configuring Parameters" was edited by mistake, WebAdmin may not run correctly.

- postgresql.conf
- pg_hba.conf

You can also modify the following files directly, but WebAdmin may not work correctly if the records span multiple lines. Therefore, change the record to a single row.

- pg_hba.conf
- pg_ident.conf

Instance configuration

1. Start WebAdmin and log in to the database server.
2. In the [Instances] tab, click .
3. Click to change the configuration.
4. Click to save your changes.

See

Select a client-side encoding system that can be converted to/from the database encoding system. Refer to "Automatic Character Set Conversion Between Server and Client" in "Server Administration" in the "PostgreSQL Documentation" for information on the encoding system combinations that can be converted.

Changing client authentication information

1. Start WebAdmin and log in to the database server.
2. In the [Instances] tab, click .

   Click to register new authentication information.

   To change authentication information, select the information, and then click .

   To delete authentication information, select the information, and then click .

Note

When creating the instance, do not delete the entry below, because it is a connection required for WebAdmin to monitor the operational status of the database:
Type= host, Database=all, User=all, and Method=md5

Editing instance information

Use the [Edit instance] page to modify the following items for an instance:

- Instance name
- Port number
- Backup storage path

1. In the [Instances] tab, click . The [Edit instance] page is displayed.
2. Modify the relevant items.

   If [Backup storage path] is changed, [Backup management] is enabled. Select the required option:

   Retain existing backup: Create a backup in [Backup storage path] and retain the existing backup in its original location.

   Copy existing backup to new path: Copy the existing backup to [Backup storage path]. A new backup will not be created. The existing backup will be retained in its original location.

   Move existing backup to new path: Move the existing backup to [Backup storage path]. A new backup will not be created.

   Remove existing backup: Create a backup in [Backup storage path]. The existing backup will be removed.
3. Click to save your changes.

Note

- The [Edit instance] page is also displayed when the user selects 'Navigate to the "Edit instance" page' from the [Anomaly Error] dialog box. Refer to "Anomaly Detection and Resolution" in the Operation Guide for information on what takes place when an anomaly is detected.
- When [Instance name] or [Instance port] is modified, the log_directory and core_directory parameters in postgresql.conf are updated. Also, the specified directories are created if they do not exist.

  Refer to “4.3.1.4 Importing Instances” for information on the format of these directories.

4.3.1.4 Importing Instances

Instances can be created using WebAdmin, or via the command line using the initdb command. Instances created using the initdb command (command line instances) can be managed using WebAdmin, however, they must first be imported into WebAdmin.

This section explains how to import command line instances into WebAdmin.

1. In the [Instances] tab, click . The [Import instance] page is displayed.
2. Enter the information for the instance being imported. Refer to "4.3.1.2 Creating an Instance" for information on the items that need to be entered.
3. Click to import the instance.

Note

- Importing neither starts nor stops the instance.

- A Windows service is automatically registered when an instance is imported into WebAdmin. If a Windows service was registered by the user prior to importing the instance, that service will not be deleted. After importing the instance into WebAdmin, it is recommended to discontinue the use of the user-created service.
- The following restrictions apply to instance import:
  - Any instance already managed by WebAdmin cannot be imported again.
  - The postgresql.conf file must be located in the same directory as [Data storage path].
  - Read/write permissions are required for [Data storage path].
  - The location specified in postgresql.conf for the following files must not have been changed:
    - hba_file
    - ident_file
  - If the following file contains records that span multiple lines, change the record to a single line before importing.
    - pg_hba.conf
    - pg_ident.conf
  - If the instance is part of a cluster that is monitored by Mirroring Controller, WebAdmin will be unable to detect the Mirroring Controller settings.
  - Instances making use of Mirroring Controller functionality should not be imported, because subsequent operations on those instances may cause unexpected and undesirable side-effects.
  - It is not possible to import and operate an instance that uses a directory mounted by Network File System (NFS).
  - You must make the following changes to the parameters in postgresql.conf prior to importing the instance in WebAdmin.

Parameter	Requirements
port	The port parameter should be uncommented.

    The log_directory and core_directory parameters in postgresql.conf are updated during import. Also, the specified directories are created if they do not exist.

    The format of these directories is as follows:

    log_directory: '*userProfileFolder**localSettingsFolder*\\Fujitsu\\fsep_*version**instanceNamePortNumber*\\log'

    core_directory: '*userProfileFolder**localSettingsFolder*\\Fujitsu\\fsep_*version**instanceNamePortNumber*\\core'

    version: product version_WA_architecture

    PortNumber: port number specified when creating the instance

    Examples:

    log_directory: 'C:\\Users\\naomi\\AppData\\Local\\Fujitsu\\fsep_150_WA_64\\myinst27599\\log'

    core_directory: 'C:\\Users\\naomi\\AppData\\Local\\Fujitsu\\fsep_150_WA_64\\myinst27599\\core'
  - When a standby instance is imported, a valid entry, using the IP address of the standby instance, must exist in the pg_hba.conf file of the master instance to allow the standby instance to connect to the master instance.
  - When a standby instance is imported, the value for "host" in the primary_conninfo parameter of postgresql.auto.conf should match the host name of the master instance.
  - When a standby instance is imported, you cannot specify "passfile" in the primary_conninfo parameter of postgresql.auto.conf. Be sure to specify "password".
  - Instances created by other operating systems cannot be imported.
  - If a Fujitsu Enterprise Postgres 10 or later instance is being imported while it is running, WebAdmin will encrypt the superuser password using SCRAM-SHA-256 authentication.

4.3.2 Using the initdb Command

This section describes the procedure to create an instance using the initdb command.

Note

If a port is blocked (access permissions have not been granted) by a firewall, enable use of the port by granting access. Refer to the vendor document for information on how to grant port access permissions.
Consider the security risks carefully when opening ports.

4.3.2.1 Creating an Instance

Create an instance, with the database cluster storage destination specified in the PGDATA environment variable or in the -D option. Furthermore, the user that executed the initdb command becomes the instance administrator.

Note

- Instances created using the initdb command (command line instances) can be managed using WebAdmin, however, they must first be imported into WebAdmin. Refer to "4.3.1.4 Importing Instances" for details.
- If creating multiple instances, ensure that there is no duplication of port numbers or the directories that store database clusters.

See

Refer to "initdb" in "Reference" in the PostgreSQL Documentation for information on the initdb command.

The procedure to create an instance is described below.

1. Use the OS user account that you want as the instance administrator.

   Connect with the server using the OS user account that you want as the instance administrator.

2. Configure the environment variables

   Configure the environment variables in the server with the newly created instance.

   Set the following environment variables:

   - PATH environment variables

     Add installDir\bin and installDir\lib.

   Example

   The following is a setting example for environment variables in which "C:\Program Files\Fujitsu\fsepv<*x*>server64" is used as the installation folder:

   Note that "<*x*>" indicates the product version.

   ```
   > SET PATH=C:\Program Files\Fujitsu\fsepv<x>server64\bin;C:\Program Files\Fujitsu
   \fsepv<x>server64\lib;%PATH%
   ```

3. Create a database cluster

   Create the database cluster with the initdb command, specifying the storage destination directory.

   Specify the transaction log storage destination and the locale setting option as required.

   Example

   ```
   > initdb -D D:\database\inst1 --waldir=E:\transaction\inst1 --lc-collate="C" --lc-ctype="C" --
   encoding=UTF8
   ```

Point

In some features, instance names are requested, and those names are required to uniquely identify the instance within the system. These features allow names that conform to WebAdmin naming conventions, so refer to the following points when determining the names:

- Maximum of 16 characters
- The first character must be ASCII alphabetic character
- The other characters must be ASCII alphanumeric characters

Note

- To balance I/O load, consider deploying the transaction log storage destination to a disk device other than the database cluster storage destination and the backup data storage destination.
- Messages may not display correctly if a value other than "C" is specified as the display language for messages.
- Specify "C" for collation and character category. Performance deteriorates if you specify a value other than "C" , although the behavior will follow the rules for particular languages, countries and regions. Furthermore, this may need to be revised when running applications on systems with different locales.
  For example, specify as follows:

  ```
  initdb --locale="C" --lc-messages="C"
  initdb --lc-collate="C" --lc-ctype="C"
  ```

- Specify an encoding system other than SQL_ASCII for the database. If SQL_ASCII is used, there is no guarantee that the encryption system for data in the database will be consistent, depending on the application used to insert the data.

See

Refer to "Locale Support" in "Localization" in "Server Administration" in the PostgreSQL Documentation for information on locales.

4. Set port number.

   Specify a port number in the port parameter of postgresql.conf. Ensure that the specified port number is not already used for other software. If a port number is not specified, "27500" is selected.

   Register the specified port numbers in the C:\Windows\System32\drivers\etc\services file if WebAdmin is used to create other instances. WebAdmin uses the services file to check if port numbers specified as available candidates have been duplicated.

   Register any name as the service name.

   ### Note

   Make a note of the port number for use in the Windows firewall settings.

5. Set the corefile output destination.

   Specify the output destination of the corefile, which can later be used to collect information for investigation, by setting the core_directory and core_contents parameters of postgresql.conf.

   ### See

   Refer to "Parameters" in the Operation Guide for information on the settings for these parameters.

6. Set the backup storage destination.

   Specify the backup data storage destination and other backup settings when backup is to be performed as a provision against database errors.

   See

   Refer to "Backup Methods" in the Operation Guide for information on specifying backup settings.

7. Register an instance in the Windows service

   Use the register mode of the pg_ctl command to register an instance in the Windows service.

   Specify the service name, user name, password and path to the instance in the pg_ctl command, and register the instance in the Windows service.

   Example

   The following is a setting example, in which the service name to register is "inst1", the user name is "fepuser", and the storage destination directory of the database cluster is "D:\database\inst1":

   ```
   > pg_ctl register -N "inst1" -U fepuser -P ******** -D D:\database\inst1
   ```

   Note

   - This command must be executed by an instance administrator user with administrator privileges. Execute the command from the [Administrator: Command Prompt] window. Right-click [Command Prompt], and then select [Run as administrator] from the menu to display the [Administrator: Command Prompt] window.
   - For the following reasons, a user name and password must always be specified:

     Because the Windows service is started up by the Network Service account, all user resources are created as resources of that account. This can result in error events such as failing to access database resources and not being able to perform backups/recovery.
     Note that if not specifying a user name and password for security reasons, you should specify the account from the Windows services list immediately after registering the instance in Windows services.
   - When entering the password that is specified in the pg_ctl command, for security reasons, you should be careful not to allow other users to access it.

   Commands such as sc query can be used to check the registration status.

8. Start an instance

   Use the following procedure to start the service:

   a. Display the [Services] window.

      In Windows, select [Administrative Tools], and then click [Services].

   b. Start the service

      From the services list, select the instance name that you wish to start, and click [Start Service].

If using commands to start the service, specify the service name using either the net start command or sc start command from the command prompt.

4.4 Configuring Remote Connections

This section describes the settings required when connecting remotely to Fujitsu Enterprise Postgres from a database application or a client command.

4.4.1 When an Instance was Created with WebAdmin

Settings related to connection

The default is to accept connections from remote computers to the database.

Change "listen_addresses" in postgresql.conf to modify the default behavior.

Refer to "Appendix D Configuring Parameters" for information on postgresql.conf.

Client Authentication Information settings

The following content is set by default when WebAdmin is used to create an instance.

- Authentication of remote connections from local machines is performed.

When changing Client Authentication Information, select [Client Authentication] from [Setting], and then change the settings.

4.4.2 When an Instance was Created with the initdb Command

Connection settings

The default setting only permits local connections from the client to the database. Remote connections are not accepted.

Change "listen_addresses" in postgresql.conf to perform remote connection.

All remote connections will be allowed when changed as shown below.

Example

```
listen_addresses = '*'
```

Also, configure the parameters shown below in accordance with the applications and number of client command connections.

Parameter name	Parameter description
superuser_reserved_connections	Number of connections reserved for database maintenance, for example backup or index rebuilding. If you need to simultaneously perform a large number of processes that exceed the default value, change this value accordingly.
max_connections	Set the value as: *numberOfSimultaneousConnectionsToInstance* + superuser_reserved_connections

Client authentication information settings

When trying to connect from a client to a database, settings are required to determine whether the instance permits connections from the client - if it does, then it is possible to make settings to determine if authentication is required.

See

Refer to "The pg_hba.conf File" in "Server Administration" in the PostgreSQL Documentation for details.

4.4.3 Windows Firewall Settings

If the Windows firewall feature is to be enabled, you should enable a port number on the database server. The following explains how to enable a port number:

Windows Server(R) 2016:

1. Select [Systems and Security] from [Control Panel] and click [Windows Firewall].
2. In the [Windows Firewall] window, click [Advanced settings].

3. In the [Windows Firewall with Advanced Security] window, click [Inbound Rules] on the left side of the window.
4. Click [New Rule] on the right side of the window.
5. In the [New Inbound Rule Wizard] window, select [Port], and then click [Next].
6. Select [TCP] and [Specific local ports], then specify the Web server port number specified during the WebAdmin setup, and then click [Next].
7. Select [Allow the connection], and then click [Next].
8. Select the profiles for which this rule applies, and then click [Next].
9. In [Name], specify the desired name, and then click [Finish].
10. In the [Windows Firewall with Advanced Security] window, check if the added rule is enabled under [Inbound Rules] in the center of the window.

In cases other than the above:

1. Select [Systems and Security] from [Control Panel] and click [Windows Defender Firewall].
2. In the [Windows Defender Firewall] window, click [Advanced settings].
3. In the [Windows Defender Firewall with Advanced Security] window, click [Inbound Rules] on the left side of the window.
4. Click [New Rule] on the right side of the window.
5. In the [New Inbound Rule Wizard] window, select [Port], and then click [Next].
6. Select [TCP] and [Specific local ports], then specify the Web server port number specified during the WebAdmin setup, and then click [Next].
7. Select [Allow the connection], and then click [Next].
8. Select the profiles for which this rule applies, and then click [Next].
9. In [Name], specify the desired name, and then click [Finish].
10. In the [Windows Defender Firewall with Advanced Security] window, check if the added rule is enabled under [Inbound Rules] in the center of the window.

4.5 Other Settings

This section describes settings that are useful for operations.

4.5.1 Error Log Settings

This section explains the settings necessary to monitor errors in applications and operations, and to make discovering the causes easier.

Make error log settings only when instances are created with the initdb command.

When creating instances with WebAdmin, these settings are already made and hence do not need to be set. Furthermore, some parameters are used by WebAdmin, and if changed, may cause WebAdmin to no longer work properly. Refer to "Appendix D Configuring Parameters" for details.

Edit the following parameters in postgresql.conf:

Parameter name	Parameter description	How to enable the settings
event_source	Specify the event source name to be attached to messages, for identifying messages output to the event log when using multiple instances.	- Restart services from the Windows services window. - Use the net command or sc command to stop and start services.

Parameter name	Parameter description	How to enable the settings
logging_collector	Specify "on" to ensure that messages are output by Fujitsu Enterprise Postgres to the server log file. The server log file is created in the log directory in the database cluster.	- Restart services from the Windows services window. - Use the net command or sc command to stop and start services.
log_destination	Specify " stderr, eventlog" to output messages from Fujitsu Enterprise Postgres to the screen and either the system log or the event log.	reload option of the pg_ctl mode
log_line_prefix	Specify information to be added at the start of messages output by an instance. This information is useful for automatic monitoring of messages. You can output the SQLSTATE value, output time, executing host, application name, and user ID. Refer to "What To Log" in the PostgreSQL Documentation for details. Example: log_line_prefix = '%e: %t [%p]: [%l-1] user = %u,db = %d,remote = %r app = %a '	reload option of the pg_ctl mode

Point

- If you want fewer application errors being output to the eventlog, refer to "When To Log" and "What To Log" in the PostgreSQL Documentation for information on how to reduce the output messages.
- If you want to separate errors output from other software, refer to "Where To Log" in the PostgreSQL Documentation to change the output destination to the server log file rather than the system log.

4.5.2 Configuring Automatic Start and Stop of an Instance

You can automatically start or stop an instance when the operating system on the database server is started or stopped.

Use the following procedure to configure automatic start and stop of an instance.

Note that, if an instance is started in a failover operation, the cluster system will control the start or stop, therefore this feature should not be used. Also, when performing database multiplexing, refer to "Enabling Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" in the Cluster Operation Guide (Database Multiplexing).

When an instance was created with WebAdmin

When an instance is created with WebAdmin, the instance is registered in the Windows service and automatic start and stop is set for the instance.

To change the automatic start and stop setting for an instance, select the service for the applicable instance in the Windows services window, and in [Startup Type], select [Automatic] or [Manual].

When an instance was created with the initdb command

When the startup type of the service is set to [Manual], change it to [Automatic]. By setting the startup type to [Automatic], the service will start up automatically when the Windows(R) system is started up, and will stop automatically when the Windows(R) system is shut down.

Note

The settings should be performed by a user with administrator privileges.

Use the following procedure to switch the service:

1. Display the [Services] window.

   In Windows, select [Administrative Tools], and then click [Services].

2. Switch the startup type

   Select the Fujitsu Enterprise Postgres service name, display the [Properties] dialog box, and then switch the startup type from [Manual] to [Automatic].

The above setting can also be changed using the sc config command.

4.5.3 Settings when Using the Features Compatible with Oracle Databases

To use the features compatible with Oracle databases, create a new instance and execute the following command for the "postgres" and "template1" databases:

```
CREATE EXTENSION oracle_compatible;
```

Features compatible with Oracle databases are defined as user-defined functions in the "public" schema created by default when database clusters are created, so they can be available for all users without the need for special settings.

For this reason, ensure that "public" (without the double quotation marks) is included in the list of schema search paths specified in the search_path parameter.

There are also considerations for use the features compatible with Oracle databases. Refer to " Precautions when Using the Features Compatible with Oracle Databases" in the Application Development Guide for details.

4.6 Integration with Message-Monitoring Software

To monitor messages output by Fujitsu Enterprise Postgres using software, configure the product to monitor SQLSTATE, instead of the message text - this is because the latter may change when Fujitsu Enterprise Postgres is upgraded.

Configure Fujitsu Enterprise Postgres to output messages in a format that can be read by the message-monitoring software by specifying "%e" in the log_line_prefix parameter of postgresql.conf to output the SQLSTATE value.

A setting example is shown below - it outputs the output time, executing host, application name, and user ID, in addition to the SQLSTATE value.

Example

```
log_line_prefix = '%e: %t [%p]: [%l-1] user = %u,db = %d,remote = %r app = %a '
```

See

Refer to "What To Log" in the PostgreSQL Documentation for information on how to configure the settings.

4.7 Setting Up and Removing OSS

This section explains how to set up OSS supported by Fujitsu Enterprise Postgres.

If you want to use OSS supported by Fujitsu Enterprise Postgres, follow the setup procedure.

If you decide not to use the OSS supported by Fujitsu Enterprise Postgres, follow the removing procedure.

Information

- In this section, the applicable database that enables the features of each OSS is described as "postgres".
- Execute CREATE EXTENSION for the "template1" database also, so that each OSS can be used by default when creating a new database.

Refer to "OSS Supported by Fujitsu Enterprise Postgres" in the General Description for information on OSS other than those described below.

4.7.1 oracle_fdw

4.7.1.1 Setting Up oracle_fdw

1. Add the path of the OCI library to the environment variable. The available version of the OCI library is 11.2 or later.
   Add the installation path of the OCI library to the PATH environment variable.
2. Open a command prompt with administrator privileges and run the following command:

```
> xcopy /E "c:\Program Files\Fujitsu\fsepv<x>server64\OSS\oracle_fdw\*"
        "c:\Program Files\Fujitsu\fsepv<x>server64"
```

3. Restart Fujitsu Enterprise Postgres.
4. Execute CREATE EXTENSION for the database that will use this feature.
   Use the psql command to connect to the "postgres" database.

```
postgres=# CREATE EXTENSION oracle_fdw;
CREATE EXTENSION
```

Information

- If the OCI library is not installed on the server, install it using the Oracle client or Oracle Instant Client.
  Refer to the relevant Oracle manual for information on the installation procedure.
- If the version of the OCI library is updated, change the path of the OCI library in the PATH environment variable to the updated path.

Note

This feature cannot be used on instances created in WebAdmin. It can only be used via server commands.

4.7.1.2 Removing oracle_fdw

1. Execute DROP EXTENSION for the database that will use this feature.
   Use the psql command to connect to the "postgres" database.

```
postgres=# DROP EXTENSION oracle_fdw CASCADE;
DROP EXTENSION
```

2. Open a command prompt with administrator privileges and run the following command:

```
> del "c:\Program Files\Fujitsu\fsepv<x>server64\filesCopiedDuringSetup"
```

Information

The files copied during setup can be checked below.

```
> dir /b /s "c:\Program Files\Fujitsu\fsepv<x>server64\OSS\oracle_fdw"
```

4.7.2 pg_hint_plan

4.7.2.1 Setting Up pg_hint_plan

1. Set the postgresql.conf file parameters.
   Add "pg_hint_plan" to the "shared_preload_libraries" parameter.

2. Open a command prompt with administrator privileges and run the following command:

```
> xcopy /E "c:\Program Files\Fujitsu\fsepv<x>server64\OSS\pg_hint_plan\*"
       "c:\Program Files\Fujitsu\fsepv<x>server64"
```

3. Restart Fujitsu Enterprise Postgres.

4. Run CREATE EXTENSION for the database that uses this feature.
   The target database is described as "postgres" here.
   Use the psql command to connect to the "postgres" database.

```
postgres=# CREATE EXTENSION pg_hint_plan;
CREATE EXTENSION
```

See

Refer to " Optimizer Hints" in the Application Development Guide for details.

4.7.2.2 Removing pg_hint_plan

Note

Unsetting pg_hint_plan will cause hints registered in the hint_plan.hints table to be lost. Therefore, it is recommended that pg_dump back up the hint_plan.hints table for each database if it is likely that pg_hint_plan will be used again later.

1. Execute DROP EXTENSION for the database that will use this feature.
   Use the psql command to connect to the "postgres" database.

```
postgres=# DROP EXTENSION pg_hint_plan CASCADE;
DROP EXTENSION
```

2. Open a command prompt with administrator privileges and run the following command:

```
> del "c:\Program Files\Fujitsu\fsepv<x>server64\filesCopiedDuringSetup"
```

Information

The files copied during setup can be checked below.

```
> dir /b /s "c:\Program Files\Fujitsu\fsepv<x>server64\OSS\pg_hint_plan"
```

3. Set the postgresql.conf file parameters.
   Delete "pg_hint_plan" to the shared_preload_libraries parameter.

4. Restart Fujitsu Enterprise Postgres.

4.7.3 pg_dbms_stats

4.7.3.1 Setting Up pg_dbms_stats

1. Set the postgresql.conf file parameter.
   Add "pg_dbms_stats" to the "shared_preload_libraries" parameter.

2. Open a command prompt with administrator privileges and run the following command:

```
> xcopy /E "c:\Program Files\Fujitsu\fsepv<x>server64\OSS\pg_dbms_stats\*"
       "c:\Program Files\Fujitsu\fsepv<x>server64"
```

3. Restart Fujitsu Enterprise Postgres.

4. Run CREATE EXTENSION for the database that will use this feature.
   The target database is described as "postgres" here.
   Use the psql command to connect to the "postgres" database.

```
postgres=# CREATE EXTENSION pg_dbms_stats;
CREATE EXTENSION
```

See

Refer to "Optimizer Hints" in the Application Development Guide for details.

4.7.3.2 Removing pg_dbms_stats

Note

Unsetting pg_dbms_stats causes statistics managed by pg_dbms_stats to be lost. Therefore, it is recommended that you back up each table in the dbms_stats folder of each database in binary format if you may want to use pg_dbms_stats again later.

```
postgres > # COPY <dbms_stats Schema's table name> TO '<Filename>' FORMAT binary;
```

1. Execute DROP EXTENSION for the database that will use this feature.
   Use the psql command to connect to the "postgres" database.

```
postgres=# DROP EXTENSION pg_dbms_stats CASCADE;
DROP EXTENSION
```

2. Open a command prompt with administrator privileges and run the following command:

```
> del "c:\Program Files\Fujitsu\fsepv<x>server64\filesCopiedDuringSetup"
```

Information

The files copied during setup can be checked below.

```
> dir /b /s "c:\Program Files\Fujitsu\fsepv<x>server64\OSS\pg_dbms_stats"
```

3. Set the postgresql.conf file parameters.
   Delete "pg_dbms_stats" to the shared_preload_libraries parameter.

4. Restart Fujitsu Enterprise Postgres.

4.8 Deleting Instances

This section explains how to delete an instance.

- 4.8.1 Using WebAdmin
- 4.8.2 Using Server Commands

Note

- Always use WebAdmin to delete instances that were created or imported using WebAdmin. Because WebAdmin management information cannot be deleted, WebAdmin will determine that the instance is abnormal.

4.8.1 Using WebAdmin

This section explains how to delete an instance using WebAdmin.

Use the following procedure to delete an instance.

1. Stop the instance

   In the [Instances] tab, select the instance to stop and click .

2. Back up files.

   Before deleting the instance, back up any required files under the data storage destination, the backup data storage destination, and the transaction log storage destination.

3. Delete the instance

   In the [Instances] tab, select the instance to delete and click .

Note

Deleting an instance deletes only the following lowest-level directories. If they are not required, delete them manually.

- Data storage destination
- Backup data storage destination
- Transaction log storage destination (if different from the data storage destination)

4.8.2 Using Server Commands

This section explains how to delete an instance using server commands.

Use the following procedure to delete an instance.

1. Stop the instance

   Stop the instance in Windows services, or use the pg_ctl command stop mode.

   Use the following procedure to stop a service in Windows services:

   a. Display the [Services] window.

      In Windows, select [Administrative Tools], and then click [Services].

   b. Stop the service

      Select the instance name that you wish to stop from the services list, and click [Stop Service]. If you stop a service while applications and commands are running, Fujitsu Enterprise Postgres will force those applications and commands to close and will stop normally.

   You can also stop a service by specifying the service name in the net stop command or sc stop command.

2. Back up files

   Before deleting the instance, you should back up all necessary files contained in the data storage destination, backup data storage destination, and transaction log storage destination.

3. Delete the instance

   Use a command such as rmdir to delete the following directories:

   - Data storage destination directory
   - Backup data storage destination directory
   - Transaction log storage destination directory (if a different directory to the data storage destination directory was specified)

Chapter 5 Uninstallation

This chapter explains the uninstallation of Fujitsu Enterprise Postgres.

Note that "*x* SP*z*" indicates the version and level of the installed product and "<*x*>" in paths indicates the product version.

Information

- If a [User Account Control] dialog box is displayed at the start of the uninstallation, click [Yes] to continue processing.

  If [No] is clicked, permission to continue is denied and an [Error] dialog box will be displayed.
  To continue the uninstallation, click [Retry] in the [Error] dialog box. To end the operation, click [Cancel].

- If uninstallation is suspended or processing terminates abnormally, the [Program Compatibility Assistant] dialog box may be displayed. Click [This program uninstalled correctly] and continue operation.

5.1 Uninstallation in Interactive Mode

The uninstallation procedure is described below.

Note

- Uninstalling removes all files and directories under the installation directory.If you have placed user files in the installation directory, you may need to save them before uninstalling.

- If performing operation with WebAdmin, back up the following folder before uninstallation.
  Instances will not be recognized by WebAdmin even if Fujitsu Enterprise Postgres is reinstalled after uninstallation.
  If performing operation with WebAdmin after reinstalling Fujitsu Enterprise Postgres, replace the following backed up file after installation.

  Follow the procedure below to perform the backup.

  1. Stop the WebAdmin server. Refer to "B.1.3 Stopping the Web Server Feature of WebAdmin" for details.
  2. Back up the following folder:

     webAdminInstallFolder\data\fepwa

  By replacing the above folder in the installation folder after installation, the instance will be recognized by WebAdmin, and the recognized instance will be set to automatically start and stop.
  To disable the automatic start and stop setting for an instance, select the service for the applicable instance in the Windows services window, and in [Startup Type], select [Manual].

- If using database multiplexing mode, refer to "Uninstalling in Database Multiplexing Mode" in the Fujitsu Enterprise Postgres Cluster Operation Guide (Database Multiplexing) before performing the uninstallation.

See

Refer to the Installation and Setup Guide for Client when uninstalling the Fujitsu Enterprise Postgres client feature.

Information

If an error occurs while the product is being uninstalled, refer to "Uninstall (middleware) Messages" in the Fujitsu Enterprise Postgres product website, and take the required action.

1. Stop applications and programs

Before starting the uninstallation, stop the following:

- Applications that use the product
- Connection Manager
- pgAdmin

2. Stop instances

Stop all instances that are using the product to be uninstalled.

Stopping of instances should be performed by the appropriate instance administrator.

When an instance was created with WebAdmin

In the [Instances] tab, select the instance to stop and click .

When an instance was created with the initdb command

Use the following procedure to stop a service:

a. Display the [Services] window

   In Windows, select [Administrative Tools], and then click [Services].

b. Stop the service

   Select the instance name that you wish to stop from the services list, and click [Stop Service]. If you stop a service while applications and commands are running, Fujitsu Enterprise Postgres will force those applications and commands to close and will stop normally.

You can also stop a service by specifying the service name in the net stop command or sc stop command.

3. Stop WebAdmin

If you are using WebAdmin, stop WebAdmin.

Refer to "B.1.3 Stopping the Web Server Feature of WebAdmin" for details.

4. Unregister Windows services

Perform this step if the instance was created with the initdb command.

Unregister the instance registered in Windows services.

Use the unregister mode of the pg_ctl command to specify the registered service name and unregister the instance from Windows services.

Example

The following is an example showing execution of this command on the registered service name "inst1".

```
> pg_ctl unregister -N "inst1"
```

Note

You should unregister services before uninstalling Fujitsu Enterprise Postgres. If you uninstall Fujitsu Enterprise Postgres while services are running, several files will remain after the uninstallation.

If you have carried out the uninstallation without unregistering services beforehand, use the server command sc delete to unregister the services.

This command must be executed by an instance administrator user with administrator privileges. Execute the command from the [Administrator: Command Prompt] window. Right-click [Command Prompt], and then select [Run as administrator] from the menu to display the [Administrator: Command Prompt] window.

5. Delete registrations related to the event log

If you are outputting to the event log, a DLL registration mentioned in "4.2.5 Preparing for Output to the Event Log" has been performed.

To prevent unnecessary issues from occurring, you should delete this registration. Refer to "Server Setup and Operation", "Registering Event Log on Windows" in the PostgreSQL Documentation for details.

The following is an example showing deletion of the DLL registration for a 64-bit product under the default event source name.

```
> regsvr32 /u "c:\Program Files\Fujitsu\fsepv<x>server64\lib\pgevent.dll"
```

If using multiple instances

DLL registration is performed so that you can output messages corresponding to the event source name assigned by the user, allowing you to identify messages output to the event log by instance.

Since it is necessary to delete the DLL registration for each instance, delete the DLL registration by event source name.

The following is an example showing deletion of the DLL of a 64-bit product registered under the event source name "Enterprise Postgres inst1".

```
> regsvr32 /u /i:"Enterprise Postgres inst1" "C:\Program Files\Fujitsu\fsepv<x>server64\lib
\pgevent.dll"
```

Note that this step is not required if the instance was created with WebAdmin.

If installing multiple versions

If the instances you created using this package have been set to output error logs to the event log, use the DLL path name that you took note of previously as explained in "4.2.5 Preparing for Output to the Event Log" to reregister the default event source name.

Ensure to delete DLLs before the uninstallation. If you perform the uninstallation without doing so, you may not be able to delete the DLLs at a later time.

6. Start Uninstall (middleware)

In Windows, click [All Programs] or [All apps], then [Fujitsu], and then [Uninstall (middleware)].

7. Select the software

Select the product to be uninstalled from [Software Name], and then click [Remove].

8. Start the uninstallation

Click [Uninstall].

9. Finish the uninstallation

The uninstallation completion window will be displayed. Click [Finish].

The installation folder may remain after uninstallation. If it is not required, delete it.

10. Stop Uninstall (middleware)

In Uninstall (middleware), click [Close].

5.2 Uninstallation in Silent Mode

The uninstallation procedure is described below.

Note

- Uninstalling removes all files and directories under the installation directory.If you have placed user files in the installation directory, you may need to save them before uninstalling.
- If performing operation with WebAdmin, back up the following folder before uninstallation.
  Instances will not be recognized by WebAdmin even if Fujitsu Enterprise Postgres is reinstalled after uninstallation.
  If performing operation with WebAdmin after reinstalling Fujitsu Enterprise Postgres, replace the following backed up file after installation.

  Follow the procedure below to perform the backup.

  1. Stop the WebAdmin server. Refer to "B.1.3 Stopping the Web Server Feature of WebAdmin" for details.
  2. Back up the following folder:

     webAdminInstallFolder\data\fepwa

  By replacing the above folder in the installation folder after installation, the instance will be recognized by WebAdmin, and the recognized instance will be set to automatically start and stop.
  To disable the automatic start and stop setting for an instance, select the service for the applicable instance in the Windows services window, and in [Startup Type], select [Manual].
- If using database multiplexing mode, refer to "Uninstalling in Database Multiplexing Mode" in the Fujitsu Enterprise Postgres Cluster Operation Guide (Database Multiplexing) before performing the uninstallation.

See

- Refer to the Installation and Setup Guide for Client when uninstalling the Fujitsu Enterprise Postgres client feature.
- Refer to the Fujitsu Enterprise Postgres product website for information on uninstallation in silent mode, such as the error messages.

1. Stop applications and programs

Before starting the uninstallation, stop the following:

- Applications that use the product
- Connection Manager
- pgAdmin

2. Stop all instances

Stop all instances that are using the product to be uninstalled.

When an instance was created with WebAdmin

In the [Instances] tab, select the instance to stop and click .

When an instance was created with the initdb command

Use the following procedure to stop a service:

a. Display the [Services] window.

   In Windows, select [Administrative Tools], and then click [Services].

b. Stop the service

   Select the instance name that you wish to stop from the services list, and click [Stop Service]. If you stop a service while applications and commands are running, Fujitsu Enterprise Postgres will force those applications and commands to close and will stop normally.

You can also stop a service by specifying the service name in the net stop command or sc stop command.

3. Stop WebAdmin

If you are using WebAdmin, stop WebAdmin.

Refer to "B.1.3 Stopping the Web Server Feature of WebAdmin" for details.

4. Unregister Windows services

Perform this step if the instance was created with the initdb command.

Unregister the instance registered in Windows services.

Use the unregister mode of the pg_ctl command to specify the registered service name and unregister the instance from Windows services.

Example

The following is an example showing execution of this command for the registered service name "inst1".

```
> pg_ctl unregister -N "inst1"
```

Note

You should unregister services before uninstalling Fujitsu Enterprise Postgres. If you uninstall Fujitsu Enterprise Postgres while services are running, several files will remain after the uninstallation.

If you have carried out the uninstallation without unregistering services beforehand, use the server command sc delete to unregister the services.

This command must be executed by an instance administrator user with administrator privileges. Execute the command from the [Administrator: Command Prompt] window. Right-click [Command Prompt], and then select [Run as administrator] from the menu to display the [Administrator: Command Prompt] window.

5. Delete registrations related to the event log

If you are outputting to the event log, a DLL registration mentioned in "4.2.5 Preparing for Output to the Event Log" has been performed.

To prevent unnecessary issues from occurring, you should delete this registration. Refer to "Server Setup and Operation", "Registering Event Log on Windows" in the PostgreSQL Documentation for details.

The following is an example showing deletion of the DLL registration for a 64-bit product under the default event source name.

```
> regsvr32 /u "c:\Program Files\Fujitsu\fsepv<x>server64\lib\pgevent.dll"
```

If using multiple instances

DLL registration is performed so that you can output messages corresponding to the event source name assigned by the user, allowing you to identify messages output to the event log by instance.

Since it is necessary to delete the DLL registration for each instance, delete the DLL registration by event source name.

The following is an example showing deletion of the DLL of a 64-bit product registered under the event source name "Enterprise Postgres inst1".

```
> regsvr32 /u /i:"Enterprise Postgres inst1" "c:\Program Files\Fujitsu\fsepv<x>server64\lib
\pgevent.dll"
```

Note that this step is not required if the instance was created with WebAdmin.

If installing multiple versions

If the instances you created using this package have been set to output error logs to the event log, use the DLL path name that you took note of previously as explained in "4.2.5 Preparing for Output to the Event Log" to reregister the default event source name.

Note

Ensure to delete DLLs before the uninstallation. If you perform the uninstallation without doing so, you may not be able to delete the DLLs at a later time.

6. Start the command prompt

In Windows, right-click [Command Prompt] and then select [Run as administrator].

7. Run the uninstaller

Execute the command below.

The installation folder may remain after uninstallation. If it is not required, delete it.

Example

```
X:> installFolder\suninst.bat
```

X: Drive on which the product is installed

Appendix A Recommended WebAdmin Environments

This appendix describes the recommended WebAdmin environment. The following explanation is based on the assumption that Microsoft Edge is used unless otherwise stated.

Note

The displayed screen varies depending on your environment, so check and set according to the screen.

A.1 Recommended Browser Settings

- Use a display resolution of 1280 x 768 or higher, and 256 colors or more.
- Select [Setting] >>[Appearance] >> [Font size] >> [Medium (Recommended)].
- Select [Setting] >>[Appearance] >> [Zoom] >> [100%].

A.2 How to Set Up the Pop-up Blocker

If the Pop-up Blocker is enabled, use the procedure below to configure settings to allow pop-ups from the server where Fujitsu Enterprise Postgres is installed.

1. Click [Setting] >> [Cookie and site permissions] >> [All Permissions] >> [Pop-ups and redirects].
   If the [Block (Recommended)] switch is not on (blue), the pop-up blocker is not working, and no further action is required.
2. Under [Pop-ups and Redirects], click the [Allow] >> [Add] button.
3. In [Add Site], in [Site], enter the address of the server where you installed Fujitsu Enterprise Postgres and click the [Add] button.
4. Close Microsoft Edge.

Appendix B Setting Up and Removing WebAdmin

This appendix describes how to set up and remove WebAdmin.

Note that "<*x*>" in paths indicates the product version.

B.1 Setting Up WebAdmin

This section explains how to set up WebAdmin.

B.1.1 Setting Up WebAdmin

Follow the procedure below to set up WebAdmin.

1. Log in

   Log in as a user that belongs to the Administrators group.

2. Display the setup window

   In Windows, click [All Programs] or [All apps], then [Product name], and then [WebAdmin Setup].

   Note

   If the same [User Account Control] dialog box as that shown below is displayed, click [Yes] to continue processing.

3. Specify the port number

   Specify the following port numbers to be used in WebAdmin.

   Refer to the services file. Only change to a different port number if the same port number is being used by another service.

   Make a note of the Web server port number, because it will be required for starting the WebAdmin window.

Item	Value (recommended value)
Web server port number	27515
WebAdmin internal port number	27516
WebAdmin automatic start	Selected

   Web server port number

   Specify an unused port number in the following range for the port number used for communication between the Web browser and Web server:

   - 1024 to 49151

   The Web server port number is registered as a port number of the following service name in the services file.

   fep_150_WA_64_WebAdmin_Port1

   WebAdmin internal port number

   Specify an unused port number in the following range for the port number used for communication between the Web server and WebAdmin runtime environment:

   - 1024 to 49151

   The WebAdmin internal port number is registered as a port number of the following service name in the services file.

   fep_150_WA_64_WebAdmin_Port2

   WebAdmin automatic start

   Select whether or not to start WebAdmin when the machine is started.

Note

- Make a note of the Web server port number for use in the Windows firewall settings.
- Unused port numbers
  In the operating system and other products, regardless of the information in the service file, unused port numbers may be automatically numbered and then used, or port numbers specified in environment files within products may also be used. Check the port numbers used by the OS and other products, and ensure that these are not duplicated.
- Access restrictions
  Prevent unauthorized access and maintain security by using a firewall product, or the packet filtering feature of a router device, to restrict access to the server IP address and the various specified port numbers.
- Port access permissions
  If a port is blocked (access permissions have not been granted) by a firewall, enable use of the port by granting access. Refer to the vendor document for information on how to grant port access permissions.
  Consider the security risks carefully when opening ports.
- Changing port numbers
  When using WebAdmin in multiserver mode, it is recommended not to change WebAdmin ports after creating instances. Otherwise, the created instances may not be accessible through WebAdmin after the port is changed.

4. Prepare for setup

Click [OK] in the setup window, and after completing the WebAdmin setup, refer to "4.2 Preparations for Setup" and perform the required preparations for setting up Fujitsu Enterprise Postgres if using WebAdmin for operation.

B.1.2 Starting the Web Server Feature of WebAdmin

Follow the procedure below to start the Web server feature of WebAdmin.

1. Display the Services window

   In Windows, select [Administrative Tools], and then click [Services].

2. Start the service

   Select the displayed name "Fujitsu Enterprise Postgres WebAdmin *version*", and then click [Start Service].

   You can also start the service by specifying the service name of the Web server feature of WebAdmin in the net start command or sc start command.

B.1.3 Stopping the Web Server Feature of WebAdmin

Follow the procedure below to stop the Web server feature of WebAdmin.

1. Display the Services window

   In Windows, select [Administrative Tools], and then click [Services].

2. Stop the service

   Select the displayed name "Fujitsu Enterprise Postgres WebAdmin *version*", and then click [Stop Service].

   You can also stop the service by specifying the service name of the Web server feature of WebAdmin in the net stop command or sc stop command.

Note

- For efficient operation of WebAdmin, it is recommended to stop the Web server feature only during a scheduled maintenance period.
- When WebAdmin is used to create and manage instances in a multiserver configuration, the Web server feature must be started and running on all servers at the same time.

B.2 Removing WebAdmin

This section explains how to remove WebAdmin.

This removal procedure stops WebAdmin and ensures that it no longer starts automatically when the machine is restarted.

To remove the setup, execute the command shown below.

Example

When WebAdmin is installed in "C:\Program Files\Fujitsu\fsepv<*x*>webadmin":

```
> C:
> cd C:\Program Files\Fujitsu\fsepv<x>webadmin\sbin
> WebAdminSetup --delete
```

Note

- The removal of the WebAdmin setup must be performed by a user with administrator privileges (a user ID that belongs to the Administrators group).
- Commands that require administrator privileges must be executed from the [Administrator: Command Prompt] window. Right-click [Command Prompt], and then select [Run as administrator] from the menu to display the [Administrator: Command Prompt] window.

B.3 Using an External Repository for WebAdmin

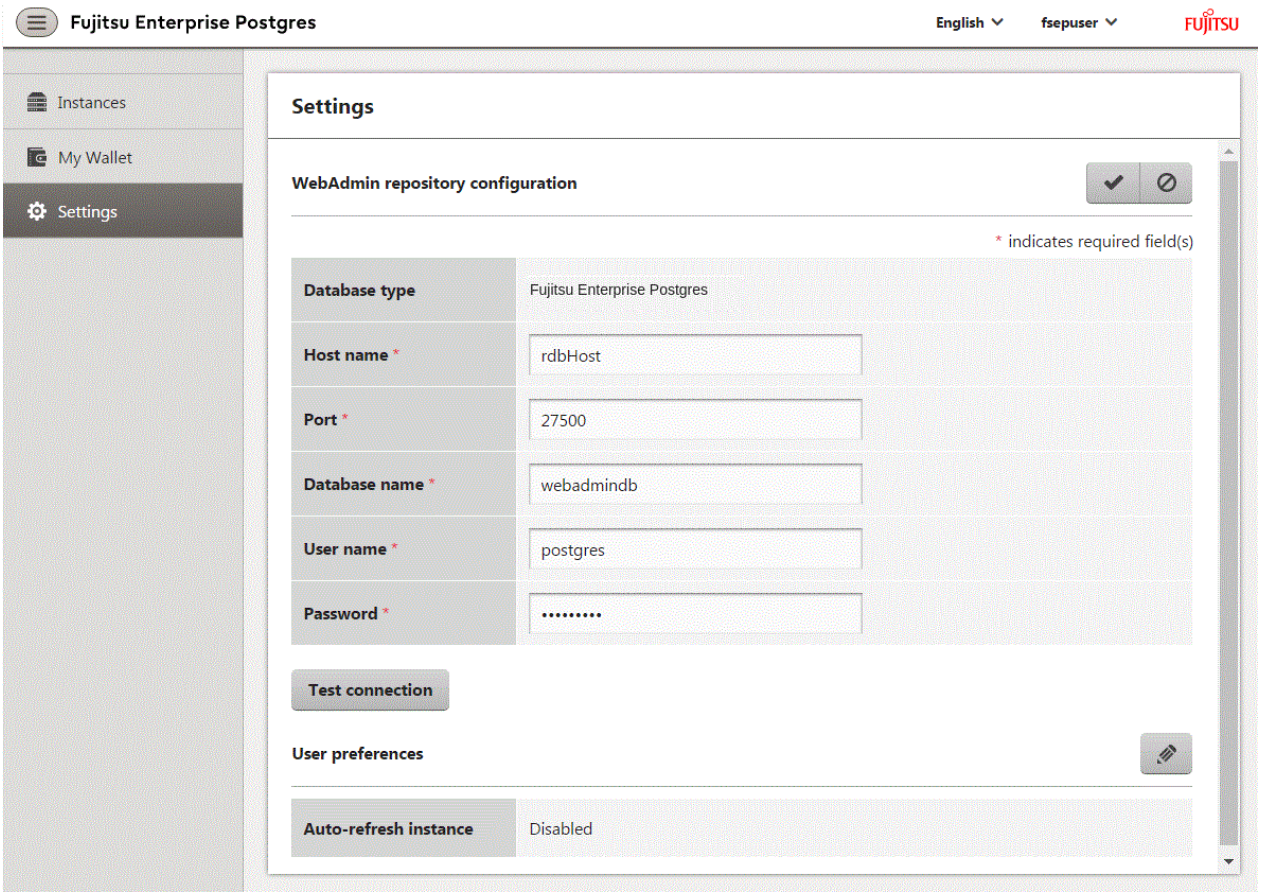
WebAdmin can be configured to use an external database, where it can store the various metadata information it uses. WebAdmin will use this database as a repository to store the information it uses to manage all the created instances. This can be a Fujitsu Enterprise Postgres database or an Open Source PostgreSQL V9.2 or later database.

Using an external database as a WebAdmin repository provides you with more flexibility in managing WebAdmin. This repository can be managed, backed up and restored as needed using pgAdmin or command line tools, allowing users to have greater flexibility and control.

Follow the procedure below to set up the repository.

1. Start WebAdmin, and log in to the database server.

2. Click the [Settings] tab, and then click in the [WebAdmin repository configuration] section.



Enter the following items:

- [Host name]: Host name of the database server
- [Port]: Port number of the database server
- [Database name]: Name of the database
- [User name]: User name to access the database
- [Password]: Password of the database user

Note

- Database type
  It is recommended to use a Fujitsu Enterprise Postgres database as a repository. A compatible PostgreSQL database can also be used as an alternative.
- It is recommended to click [Test connection] to ensure that the details entered are valid and WebAdmin is able to connect to the target database.
- Host name, Database name, User name, Password should not contain hazardous characters. Refer to "Appendix C WebAdmin Disallow User Inputs Containing Hazardous Characters".

3. Click to register the repository details.

Note

- Once the repository is set up, it can be changed any number of times by the user logged into WebAdmin. When a repository is changed:
  - It is recommended to preload the backup into this database.
  - If the data is not preloaded, WebAdmin will create a new repository.
- The database repository can be set up even after WebAdmin was already used to create instances. In that scenario, the instances already created are retained and can continue to be operated on.
- If the instance used as a repository is stopped, WebAdmin will be unusable. For this reason, it is recommended to be familiar with starting an instance from the command line. If the instance is stopped for any reason, start it from the command line and WebAdmin will be usable again.

B.4 Using the WebAdmin Auto-Refresh Feature

The WebAdmin auto-refresh feature automatically refreshes the operating status of all instances in the Instance list at the specified interval. It also refreshes the details of the selected instance.

Follow the procedure below to configure the auto-refresh options.

1. Click the [Settings] tab, and then click in the [User preferences] section.
2. Enter the following items:
   - [Auto-refresh instance]: To use the auto-refresh feature, select "Enabled". The default is "Disabled".
   - [Refresh interval (seconds)]: Number of seconds between each refresh. This is a countdown timer, which is reset every time the instance status is refreshed by any operation. Specify a value from 30 to 3600 (seconds). The default is 30.
3. Click to save the auto-refresh settings.

Note

- Auto-refresh will run only if the [Instances] page is displayed and no user-initiated operation is in progress.
- A text indicator, which is independent of auto-refresh, is displayed at the top of the Instance list. It is dynamically updated to display when the page was last refreshed.

Appendix C WebAdmin Disallow User Inputs Containing Hazardous Characters

WebAdmin considers the following as hazardous characters, which are not allowed in user inputs.

```
| (pipe sign)

& (ampersand sign)

; (semicolon sign)

$ (dollar sign)

% (percent sign)

@ (at sign)

' (single apostrophe)

" (quotation mark)

\' (backslash-escaped apostrophe)

\" (backslash-escaped quotation mark)

<> (triangular parenthesis)

() (parenthesis)

+ (plus sign)

CR (Carriage return, ASCII 0x0d)

LF (Line feed, ASCII 0x0a)

, (comma sign)

\ (backslash)
```

Appendix D Configuring Parameters

WebAdmin operates and manages databases according to the contents of the following configuration files:

- postgresql.conf

  Contains various items of information that define the operating environment of Fujitsu Enterprise Postgres.

- pg_hba.conf

  Contains various items of information related to client authentication.

These configuration files are deployed to a data storage destination. Data is written to them when the instance is created by WebAdmin and when settings are changed, and data is read from them when the instance is started and when information from the [Setting] menu is displayed.

Direct editing of each configuration file is possible with a text editor.

See

Refer to "Server Configuration" and "Client Authentication" in "Server Administration" in the PostgreSQL Documentation for information on the parameters.

Note

WebAdmin checks for port number and backup storage path anomalies when various operations are performed. An anomaly occurs when the value of [Port number] and/or [Backup storage path] in WebAdmin is different from the value of the corresponding parameter in postgresql.conf. Refer to "Anomaly Detection and Resolution" in the Operation Guide for details.

postgresql.conf

Parameters that can be changed in WebAdmin

The postgresql.conf parameters that can be changed in WebAdmin are shown below:

Section	WebAdmin item	postgresql.conf file parameter
Instance Configuration		
Character encoding	Character set	client_encoding
	Message locale	lc_messages
Communication	Max connections	max_connections
SQL options	Transform NULL format	transform_null_equals
	Date output format	DateStyle (*1)
	Interval output format	IntervalStyle
	Number of digits for floating values	extra_float_digits
	Transaction isolation levels	default_transaction_isolation
	Currency format	lc_monetary
	Date and time format	lc_time
	Numerical value format	lc_numeric
Memory	Sort memory (KB)	work_mem
	Shared buffers (KB)	shared_buffers
Streaming replication	WAL level	wal_level

Section	WebAdmin item	postgresql.conf file parameter
	Maximum WAL senders	max_wal_senders
	WAL save size (MB)	wal_keep_size
	Hot standby	hot_standby
	Synchronous standby names	synchronous_standby_names
	WAL receiver timeout (ms)	wal_receiver_timeout
Edit instance		
	Instance name	n/a
	Instance port	port
	Backup storage path	backup_destination

*1: If you specify "Postgres" as the output format, dates will be output in the "12-17-1997" format, not the "Wed Dec 17 1997" format used in the PostgreSQL Documentation.

Information

- Calculate the maximum number of connections using the formula below:

```
maximumNumberOfConnections = maximumNumberOfConnectionsFromApplications + 3 (*1)
```

*1: 3 is the default number of connections required by the system.
Calculate the maximum number of connections using the following formula when changing superuser_reserved_connections (connections reserved for use by the superuser) in postgresql.conf.

```
maximumNumberOfConnections = maximumNumberOfConnectionsFromApplications +
superuser_reserved_connections
```

- Also check if the memory used exceeds the memory installed (refer to "Parameters automatically set by WebAdmin according to the amount of memory").

Parameters set by WebAdmin

The following postgresql.conf parameters are set by WebAdmin during instance startup (they will be ignored even if specified in postgresql.conf):

Parameter	Value
event_source (*1)	'fsep_*version_userName_instanceNamePortNumber*'
listen_addresses	*
log_destination	'stderr,eventlog'
logging_collector	on
log_line_prefix	'%e: %t [%p]: [%l-1] user = %u,db = %d,remote = %r app = %a '
log_filename (*2) (*3)	'logfile-%a.log'
log_truncate_on_rotation	on
log_rotation_age	1d

*1: *PortNumber* is the port number of the database server specified when creating the instance.

*2: The server logs are split into files based on the day of the week, and are rotated after each week.

*3: If the date changes while the instance is stopped, old logs are not deleted and continue to exist.
Manually delete old logs that are no longer required to release disk space.

Parameters automatically set by WebAdmin according to the amount of memory

The postgresql.conf parameters automatically set according to the amount of installed memory, during the creation of instances by WebAdmin, are shown below:

Parameter	Value
shared_buffers	30% of the machine's installed memory
work_mem	30% of the machine's installed memory / max_connections / 2
effective_cache_size	75% of the machine's installed memory
maintenance_work_mem	10% of the machine's installed memory / (1 + autovacuum_max_workers) (*1)

*1: The value will be capped at 2097151 KB.

When determining the values to be configured in the above parameters, you must take into account any anticipated increases in access volume or effects on performance during business operations, such as the number of applications and commands that will access the instance, and the content of processes. Also, note that in addition to Fujitsu Enterprise Postgres, other software may be running on the actual database server. You will need to determine the degree of priority for the database and other software, as well as the memory allocation size.

WebAdmin automatically configures complex parameter settings such as those mentioned above, based on the size of the internal memory of the machine. This enables maximum leverage of the machine memory to facilitate resistance against fluctuations during business operations.

Accordingly, the effects of the above-mentioned factors must be estimated and taken into account when determining and configuring parameter values, so that memory resources can be effectively allocated among other software or instances, and so that adverse effects can be mutually avoided. Refer to "Memory" in "Resource Consumption", and "Planner Cost Constants" in "Query Planning", under "Server Administration" in the PostgreSQL Documentation for information on parameter values and required considerations.

Parameter values can be modified using the WebAdmin [Setting] menu, or edited directly using a text editor.

If adding an instance, determine the parameter values, including for existing instances, and make changes accordingly.

Note

- Do not directly edit the following postgresql.conf parameters with a text editor, otherwise WebAdmin may not work properly if you make a mistake):
  - archive_mode
  - archive_command
  - wal_level
  - wal_sync_method
  - log_line_prefix
  - log_destination
  - logging_collector
  - log_directory
  - log_file_mode
  - log_filename
  - log_truncate_on_rotation
  - log_rotation_age
  - event_source

- You must take care with the following parameter:
  - superuser_reserved_connections

    Set it to a number that includes the 3 connections required in WebAdmin (the default is 3).

pg_hba.conf

Refer to "Client Authentication" in "Server Administration" in the PostgreSQL Documentation for information on content that can be configured in pg_hba.conf.

- Use the following client authentication settings to allow the instance administrator to connect to the database using WebAdmin:
  - The connection type: "host"
  - The IP address is a loopback address ("127.0.0.1/32")
- If you specify an item or value that cannot be set by WebAdmin when editing the pg_hba.conf file with a text editor, it will not be possible to reference that line from WebAdmin.

Appendix E Estimating Database Disk Space Requirements

This appendix describes how to estimate database disk space requirements.

E.1 Estimating Table Size Requirements

The following tables provide the formulas for estimating table size requirements.

Table E.1 Estimation formula when the record length is 2032 bytes or less

Item	Estimation formula (bytes)
(1) Record length	27(*1) + NULL map + OID + column data NULL map: Number of columns / 8 (*2) OID: 4 Column data: Sum of column lengths *1: Record header section *2: Round the result up to the next integer. - Because the column data is placed in boundaries of 8 bytes, you need to make an adjustment so that the sum of the record header section, NULL map and OID is a multiple of 8. For example, if the calculated length is 27 + 1 / 8 (rounded up) + 0 = 28 bytes, add 4 to make the length 32 bytes. - Because the data of each column is placed in boundaries of the defined data type, take the boundary of each data type into account for the length of the column data. For example, the length of the column data in the table below will not be the sum of the data types, which is 37 bytes, but will instead be 64 bytes following boundary adjustment. Definition: create table tb1(c1 char(1), c2 bigint, c3 int, c4 box) Estimation: CHAR type 1 byte + boundary adjustment of 7 bytes for BIGINT type 8 bytes + BIGINT type 8 bytes + INT type 4 bytes + boundary adjustment of 12 bytes for BOX type 32 bytes + BOX type 32 bytes = 64 bytes - Because each record is placed in boundaries of 8 bytes, you need to make an adjustment so that the length of the column data is a multiple of 8. - If the calculated record length exceeds 2,032 bytes, the variable length data in the record might be compressed automatically. If so, use the estimation formulas in "Table E.2 Estimation formula when the record length exceeds 2032 bytes" to estimate the table size.
(2) Page size requirement	8192 (*1) × fillfactor (*2) - 24 (*3) *1: Page length (8192) *2: Value of the fillfactor specified in the table definitions (if omitted, 100%) *3: Page header (24) - The calculated (2) page size requirement will be rounded down to the nearest integer.
(3) Number of records per page	(2) Page size requirement / ((1) record length + 4 (*1)) *1: Pointer length (4) - The result will be rounded down to the nearest integer.

Item	Estimation formula (bytes)
(4) Number of pages required for storing records	Total number of records / (3) number of records per page - The result will be rounded up to the next integer.
(5) Amount of space	(4) Number of pages required for storing records x page length x safety factor (*1) *1: Specify 2.0 or higher. - This is the safety factor assumed if vacuuming is performed for garbage collection in tables and indexes.

Table E.2 Estimation formula when the record length exceeds 2032 bytes

Item	Estimation formula (bytes)
(5) Amount of space	Total number of records x (1) record length x safety factor (*1) *1: Specify 2.0 or higher. - This is the safety factor assumed if vacuuming is performed for garbage collection in tables and indexes.

E.2 Estimating Index Size Requirements

This section provides the formulas for estimating index size requirements.

Fujitsu Enterprise Postgres provides six index types: B-tree, Hash, GiST, GIN, SP-GiST, and VCI. If you do not specify the index type in the CREATE INDEX statement, a B-tree index is generated.

The following describes how to estimate a B-tree index. Refer to "E.7 Estimating VCI Disk Space Requirements" for information on how to estimate VCI.

A B-tree index is saved as a fixed-size page of 8 KB. The page types are meta, root, leaf, internal, deleted, and empty. Since leaf pages usually account for the highest proportion of space required, you need to calculate the requirements for these only.

Table E.3 Estimation formula when the key data length is 512 bytes or less

Item	Estimation formula (bytes)
(1) Entry length	8 (*1) + key data length (*2) *1: Entry header *2: The key data length depends on its data type (refer to "E.3 Sizes of Data Types" for details). Because each entry is placed in boundaries of 8 bytes, you need to make an adjustment so that the length of the key data is a multiple of 8. For example, if the calculated length is 28 bytes, add 4 to make the length 32 bytes. - If the key data length exceeds 512 bytes, key data may be automatically compressed. In this case, use the estimation formula given in "Table E.4 Estimation formula when the key data length exceeds 512 bytes" to estimate the key data length.
(2) Page size requirement	8192 (*1) × fillfactor (*2) - 24 (*3) - 16 (*4) *1: Page length (8192) *2: Value of the fillfactor specified in the index definitions (if omitted, 90%) In the case of indexes of primary key constraints and unique constraints, the value of the fillfactor specified for each constraint in the table definitions (if omitted, 90%) *3: Page header (24) *4: Special data (16)

Item	Estimation formula (bytes)
	- The calculated (2) page size requirement will be rounded down to the nearest integer.
(3) Number of entries per page	(2) Page size requirement / ((1) entry length + 4 (*1)) *1: Pointer length - Result of (3) number of entries per page will be rounded down to the nearest integer.
(4) Number of pages required for storing indexes	Total number of records / (3) number of entries per page - Result of (4) number of pages required for storing indexes will be rounded up to the nearest integer.
(5) Space requirement	(4) Number of pages required for storing indexes x 8192 (*1) / usage rate (*2) *1: Page length *2: Specify 0.7 or lower.

Table E.4 Estimation formula when the key data length exceeds 512 bytes

Item	Estimation formula (bytes)
(5) Space requirement	Total number of records x key data length x compression ratio (*1) / usage rate (*2) *1: The compression ratio depends on the data value, so specify 1. *2: Specify 0.7 or lower as the usage rate.

E.3 Sizes of Data Types

This section lists the sizes of the data types.

E.3.1 Sizes of Fixed-Length Data Types

The following table lists the sizes of fixed-length data types.

Data type	Size (bytes)
SMALLINT (INT2)	2
INTEGER (INT4)	4
BIGINT (INT8)	8
REAL	4
DOUBLE PRECISION	8
SERIAL (SERIAL4)	4
BIGSERIAL (SERIAL8)	8
MONEY	8
FLOAT	8
FLOAT (1-24)	4
FLOAT (25-53)	8
TIMESTAMP WITHOUT TIME ZONE	8
TIMESTAMP WITH TIME ZONE	8
DATE	4
TIME WITHOUT TIME ZONE	8
TIME WITH TIME ZONE	12

Data type	Size (bytes)
INTERVAL	12
BOOLEAN	1
CIDR	IPv4: 7 IPv6: 19
INET	IPv4: 7 IPv6: 19
MACADDR	6
MACADDR8	8
POINT	16
LINE	32
LSEG	32
BOX	32
CIRCLE	24

E.3.2 Sizes of Variable-Length Data Types

The following table lists the sizes of variable-length data types.

Data type	Size (bytes)	Remarks
path	Length of size portion + 12 + 16 x number of vertices	1) When carrying out division, round to the next integer. 2) If the real data length is less than 127, then the length of the size portion is 1 byte, otherwise it is 4 bytes. 3) The number of bytes per character depends on the character set (refer to "E.3.4 Number of Bytes per Character" for details).
polygon	Length of size portion + 36 + 16 x number of vertices	
decimal	Length of size portion + 2 + (integer precision / 4 + decimal precision / 4) x 2	
numeric		
bytea	Length of size portion + real data length	
character varying(*n*), varchar(*n*)	Length of size portion + number of characters x number of bytes per character	
character(*n*), char(*n*)	Length of size portion + *n* x number of bytes per character	
text	Length of size portion + number of characters x number of bytes per character	

E.3.3 Sizes of Array Data Types

The following table lists the sizes of array data types.

Data type	Size (bytes)	Remarks
Array	Length of size portion + 12 + 8 x number of dimensions + data size of each item	If the real data length is less than 127, then the length of the size portion is 1 byte, otherwise it is 4 bytes. - Example of estimation when array data is "ARRAY[[1,2,3], [1,2,3]]" Number of dimensions: 2 INTEGER data size: 4 Total size = 1+12+8x2+6x4 = 53

E.3.4 Number of Bytes per Character

The following table lists the number of bytes per character.

The given values relate to the common character sets EUC-JP and UTF8.

Character type	Character set	Number of bytes per character
ASCII	EUC_JP	1
Halfwidth katakana	EUC_JP	2
JIS X 0208 kanji characters	EUC_JP	2
JIS X 0212 kanji characters	EUC_JP	3
ASCII	UTF8	1
Halfwidth katakana	UTF8	3
JIS X 0208 kanji characters	UTF8	3
JIS X 0212 kanji characters	UTF8	3

E.4 Estimating Transaction Log Space Requirements

This section provides the formula for estimating transaction log space requirements.

```
Transaction log space requirements = max_wal_size
```

However, if the update volume is extremely high (for example, due to a large data load and batch processing), disk writing at a checkpoint may not be able to keep up with the load, and a higher number of transaction logs than indicated here may temporarily be accumulated.

E.5 Estimating Archive Log Space Requirements

This section explains how to estimate archive log space requirements.

The archive log is an archive of the transaction logs from the time of a previous backup to the present, so it fluctuates depending on the backup period and the content of update transactions.

The longer the backup period and the more update transactions, the greater the space required for the archive log.

Therefore, measure the actual archive log space by using a test environment to simulate backup scheduling and database update in a real operating environment.

E.6 Estimating Backup Disk Space Requirements

This section provides the formula for estimating backup disk space requirements.

```
Backup disk space requirements = size of the database cluster x 2 + transaction log space requirements
                               + archive log space requirements
```

If the pgx_dmpall command performs a backup using a user exit, the backup disk size differs according to the database resources targeted for backup and the copy method.

E.7 Estimating VCI Disk Space Requirements

This section provides the formula for estimating VCI disk space requirements.

```
Disk space = (number of rows in tables) x (number of bytes per row) x (compression ratio) + (WOS size)
```

Number of bytes per row

```
Number of bytes per row = (19 + (number of columns specified in CREATE INDEX) / 8
                                  + (number of bytes per single column value)) x 1.1
```

Note: Round up the result to the nearest integer.

Compression ratio

Specify a value between 0 and 1. Since compression ratio depends on the data being compressed, use actual data or test data that simulates it, then compare the value with the estimation result. As a guide, the compression ratio measured with the Fujitsu sample data is shown below:

- Data with high degree of randomness (difficult to compress): Up to approximately 0.9 times.
- Data with high degree of similarity (easy to compress): Up to approximately 0.5 times.

WOS size

```
WOS size = (number of WOS rows) / 185 x 8096
```

One row is added to the number of WOS rows for each INSERT and DELETE, and two rows are added for UPDATE. On the other hand, the number decreases to 520,000 rows or less during conversion to ROS performed by the ROS control daemon.

Note

VCI does not support retrieval of disk space usage using the database object size function pg_indexes_size. To find out the actual total VCI disk space, check the disk space of the storage directory using an OS command or other method.

Appendix F Estimating Memory Requirements

This appendix explains how to estimate the memory.

F.1 Fujitsu Enterprise Postgres Memory Requirements

This section describes the formulas for estimating Fujitsu Enterprise Postgres memory requirements.

Use the following formula to obtain a rough estimate of memory required for Fujitsu Enterprise Postgres:

```
fujitsuEnterprisePostgresRequiredMemory = sharedMemoryAmount + localMemoryAmount
```

Shared memory amount

Refer to "Shared Memory and Semaphores" under "Server Administration" in the PostgreSQL Documentation for information on shared memory. If you enable the Global Meta Cache feature, you must also add the value of pgx_global_metacache. Refer to "Parameters" in the Operation Guide for the setting values.

However, note that if instances have been created using WebAdmin, the parameters below will be configured automatically when the instances are created. Take this into account when calculating the shared memory size.

Parameter name	Set value
shared_buffers	30 percent of the internal memory of the machine.
max_connections	100
max_prepared_transactions	100

Local memory amount

```
localMemoryAmount = processStackArea
                  + memoryUsedInDbSessionsThatUseTempTables
                  + memoryUsedInDbSessionsThatPerformSortAndHashTableOperations
                  + memoryUsedInMaintenanceOperations
                  + baseMemoryUsedInEachProcess
                  + memoryUsedPreparingForDataAccess
```

Process stack area

```
processStackArea
  = max_stack_depth x (max_connections + autovacuum_max_workers + 9)
```

This formula evaluates to the maximum value.
Actually it is used according to the growth of the stack.
In the formula above, 9 is the number of processes that perform roles specific to servers.

Memory used in database sessions that use temporary tables

```
memoryUsedInDbSessionsThatUseTempTables
  = temp_buffers x max_connections
```

This formula evaluates to the maximum value.
Memory is gradually used as temporary buffers are used, and is released when the session ends.

Memory used in database sessions that perform sort and hash table operations

```
memoryUsedInDbSessionsThatPerformSortAndHashTableOperations
  = work_mem (*1) x max_connections
```

*1) For hash table operations, multiply work_mem by hash_mem_multiplier.
This formula evaluates to the maximum value.
Memory is gradually used as operations such as sort are performed, and is released when the query ends.

Memory used in maintenance operations

```
memoryUsedInMaintenanceOperations
  = maintenance_work_mem x (numOfSessionsPerformingMaintenance + autovacuum_max_workers)
```

Note that 'maintenance operations' are operations such as VACUUM, CREATE INDEX, and ALTER TABLE ADD FOREIGN KEY.

Base memory used in each process

```
baseMemoryUsedInEachProcess
  = baseMemoryUsedInOneProcess x (max_connections + autovacuum_max_workers + 9)
```

Use the result of the following formula for memory consumed per process. This formula evaluates to the memory used when server processes are running.
In the formula above, 9 is the number of processes that perform roles specific to servers.

The amount of memory consumed per process is determined by the number of tables, indexes, and all columns of all tables that the process accesses. If your system has about 100 tables, you can estimate it to be 3 MB, but otherwise use the following estimate:

```
baseMemoryUsedInOneProcess
  = (1.9KB x All user tables + 2.9KB x All user indexes + 1.0KB x All user columns) x 1.5(*1)

If you enable the Global Meta Cache feature, use the following formula:
baseMemoryUsedInOneProcess
 = (All user tables + All user indexes + All user columns) x 1.0KB x 1.5 (*1)
    + (All user tables x 1.4KB + All user indexes x 2.4KB)
```

*1) Safety Factor (1.5)
There are variable length information. This value takes that into account.

Memory used preparing for data access

```
memoryUsedPreparingForDataAccess
 = variationAmount x (max_connections + autovacuum_max_workers + 4)

  where variationAmount = shared_buffers / 8KB x 4 bytes
      (note that 8KB is the page length, and 4 bytes is the size of page management data)
```

This formula evaluates to the memory required to access the database cache in the shared memory.
In the formula above, among the processes that perform roles specific to servers, 4 is the number of processes that access the database.

F.2 Database Multiplexing Memory Requirements

This section describes the formula for estimating database multiplexing memory requirements for the database server.

Use the following formula to obtain a rough estimate of memory required for database multiplexing:

```
Memory usage of the database multiplexing feature for the database server
           = Peak memory usage of the Mirroring Controller processes
           + Peak memory usage of the Mirroring Controller commands

Peak memory usage of the Mirroring Controller processes=150 MB

Peak memory usage of the Mirroring Controller commands=50 MB x Number of commands executed
simultaneously
```

F.3 VCI Memory Requirements

This section describes the formula for estimating VCI memory requirements.

Use the following formula to obtain a rough estimate of memory requirements:

```
memUsedByVci = memForData + memForEachProcess
```

Memory required to store data in memory

Secure the space estimated using the formula below on the stable buffer (part of shared_buffers).

```
memForData = (numOfRowsInTables) x (numOfBytesPerRow) + (wosSize)
```

Number of bytes per row

```
numOfBytesPerRow
 = (19 + (numOfColsInCreateIndexStatement) / 8 + (numOfBytesPerSingleColValue)) x 1.1
```

Note: Round up the result to the nearest integer.

WOS size

```
wosSize = (numOfWosRows) / 185 x 8096
```

One row is added to the number of WOS rows for each INSERT and DELETE, and two rows are added for UPDATE. On the other hand, the number decreases to 520,000 rows or less during conversion to ROS performed by the ROS control daemon.

Memory required for each process

```
memForEachProcess
 = memUsedPerScanning
 + memUsedForVciMaintenace
 + memUsedByCreateIndexStatement
```

Memory used per scanning

- Parallel scan

```
memUsedPerScanning
 = vci.shared_work_mem + (numOfParallelWorkers + 1) x vci.maintenance_work_mem
```

Note: The number of parallel workers used by VCI simultaneously in the entire instance is equal to or less than vci.max_parallel_degree.

- Non-parallel scan

```
memUsedPerScanning = vci.max_local_ros + vci.maintenance_work_mem
```

Note

- vci.shared_work_mem, and vci.max_local_ros are used to create local ROS. If local ROS exceeds these sizes, execute a query without using VCI according to the conventional plan.
- vci.maintenance_work_mem specifies the memory size to be secured dynamically. If it exceeds the specified value, a disk temporary file is used for operation.

Memory used for VCI maintenance

```
memUsedForVciMaintenace = vci.maintenance_work_mem x vci.control_max_workers
```

Memory used by CREATE INDEX

```
memUsedByCreateIndexStatement = vci.maintenance_work_mem
```

Note

vci.maintenance_work_mem specifies the memory to be secured dynamically. If it exceeds the specified value, a disk temporary file is used for operation.

F.4 High-Speed Data Load Memory Requirements

This section describes the formula for estimating memory requirements for the high-speed data load feature.

Use the following formula to obtain a rough estimate of memory requirements:

```
Memory usage of high speed data load
 = (Peak memory usage of pgx_loader processes + Peak memory usage of the pgx_loader commands)
 x Number of commands executed simultaneously

Peak memory usage of pgx_loader processes
 = Peak memory usage of the backend process   (6 MB)
 + Peak memory usage of parallel workers       (6 MB x number of parallel workers)
 + Peak memory usage of dynamic shared memory (80 MB x number of parallel workers)

Peak memory usage of the pgx_loader commands=9 MB
```

Point

In addition to the size calculated using the formula above, the database cache on the shared memory estimated using the shared_buffers parameter is consumed according to the size of the data (table and index keys) loaded using this feature. Refer to "E.1 Estimating Table Size Requirements" and "E.2 Estimating Index Size Requirements" for information on estimating an appropriate shared buffers value.

F.5 Global Meta Cache Memory Requirements

This section describes the formula for estimating Global Meta Cache memory requirements.

The memory calculated by "Size of the GMC area" is allocated to the shared memory. The memory calculated by the per-process meta cache management information is allocated to the local memory. Refer to the graphic in "Architecture of Global Meta Cache Feature" in the "Memory usage reduction by Global Meta Cache" in the General Description for more information.

Use the following formula to obtain a rough estimate of memory requirements:

```
Amount of memory used by the Global Meta Cache feature
  = Size of GMC area + Per-process meta cache management information

Size of GMC area = (All user tables x 0.4 KB
     + All user indexes x 0.3 KB
     + All user columns x 0.8 KB) x 1.5 (*1)

Per-process meta cache management information
  = (All user tables + All user indexes + All user columns) x 0.1KB x max_connections x 1.5 (*1)
```

*1) Safety Factor (1.5)
This value takes into account the case where both GMC before and after the change temporarily exist at the same time in shared memory when the table definition is changed or the row of the system catalog is changed.

Appendix G Quantitative Limits

This appendix lists the quantitative limits of Fujitsu Enterprise Postgres.

Table G.1 Length of identifier

Item	Limit
Database name	Up to 63 bytes (*1) (*2)
Schema name	Up to 63 bytes (*1) (*2)
Table name	Up to 63 bytes (*1) (*2)
View name	Up to 63 bytes (*1) (*2)
Index name	Up to 63 bytes (*1) (*2)
Tablespace name	Up to 63 bytes (*1) (*2)
Cursor name	Up to 63 bytes (*1) (*2)
Function name	Up to 63 bytes (*1) (*2)
Aggregate function name	Up to 63 bytes (*1) (*2)
Trigger name	Up to 63 bytes (*1) (*2)
Constraint name	Up to 63 bytes (*1) (*2)
Conversion name	Up to 63 bytes (*1) (*2)
Role name	Up to 63 bytes (*1) (*2)
Cast name	Up to 63 bytes (*1) (*2)
Collation sequence name	Up to 63 bytes (*1) (*2)
Encoding method conversion name	Up to 63 bytes (*1) (*2)
Domain name	Up to 63 bytes (*1) (*2)
Extension name	Up to 63 bytes (*1) (*2)
Operator name	Up to 63 bytes (*1) (*2)
Operator class name	Up to 63 bytes (*1) (*2)
Operator family name	Up to 63 bytes (*1) (*2)
Rewrite rule name	Up to 63 bytes (*1) (*2)
Sequence name	Up to 63 bytes (*1) (*2)
Text search settings name	Up to 63 bytes (*1) (*2)
Text search dictionary name	Up to 63 bytes (*1) (*2)
Text search parser name	Up to 63 bytes (*1) (*2)
Text search template name	Up to 63 bytes (*1) (*2)
Data type name	Up to 63 bytes (*1) (*2)
Enumerator type label	Up to 63 bytes (*1) (*2)

*1: This is the character string byte length when converted by the server character set character code.

*2: If an identifier that exceeds 63 bytes in length is specified, the excess characters are truncated and it is processed.

Table G.2 Database object

Item	Limit
Number of databases	Less than 4,294,967,296 (*1)

Item	Limit
Number of schemas	Less than 4,294,967,296 (*1)
Number of tables	Less than 4,294,967,296 (*1)
Number of views	Less than 4,294,967,296 (*1)
Number of indexes	Less than 4,294,967,296 (*1)
Number of tablespaces	Less than 4,294,967,296 (*1)
Number of functions	Less than 4,294,967,296 (*1)
Number of aggregate functions	Less than 4,294,967,296 (*1)
Number of triggers	Less than 4,294,967,296 (*1)
Number of constraints	Less than 4,294,967,296 (*1)
Number of conversion	Less than 4,294,967,296 (*1)
Number of roles	Less than 4,294,967,296 (*1)
Number of casts	Less than 4,294,967,296 (*1)
Number of collation sequences	Less than 4,294,967,296 (*1)
Number of encoding method conversions	Less than 4,294,967,296 (*1)
Number of domains	Less than 4,294,967,296 (*1)
Number of extensions	Less than 4,294,967,296 (*1)
Number of operators	Less than 4,294,967,296 (*1)
Number of operator classes	Less than 4,294,967,296 (*1)
Number of operator families	Less than 4,294,967,296 (*1)
Number of rewrite rules	Less than 4,294,967,296 (*1)
Number of sequences	Less than 4,294,967,296 (*1)
Number of text search settings	Less than 4,294,967,296 (*1)
Number of text search dictionaries	Less than 4,294,967,296 (*1)
Number of text search parsers	Less than 4,294,967,296 (*1)
Number of text search templates	Less than 4,294,967,296 (*1)
Number of data types	Less than 4,294,967,296 (*1)
Number of enumerator type labels	Less than 4,294,967,296 (*1)
Number of default access privileges defined in the ALTER DEFAULT PRIVILEGES statement	Less than 4,294,967,296 (*1)
Number of large objects	Less than 4,294,967,296 (*1)
Number of index access methods	Less than 4,294,967,296 (*1)

*1: The total number of all database objects must be less than 4,294,967,296.

Table G.3 Schema element

Item	Limit
Number of columns that can be defined in one table	From 250 to 1600 (according to the data type)
Table row length	Up to 400 gigabytes
Number of columns comprising a unique constraint	Up to 32 columns
Data length comprising a unique constraint	Less than 2,000 bytes (*1) (*2)

Item	Limit
Table size	Up to 32 terabyte
Search condition character string length in a trigger definition statement	Up to 800 megabytes (*1) (*2)
Item size	Up to 1 gigabyte

*1: Operation might proceed correctly even if operations are performed with a quantity outside the limits.

*2: This is the character string byte length when converted by the server character set character code.

Table G.4 Index

Item	Limit
Number of columns comprising a key (including VCI)	Up to 32 columns
Key length (other than VCI)	Less than 2,000 bytes (*1)

*1: This is the character string byte length when converted by the server character set character code.

Table G.5 Data types and attributes that can be handled

Item			Limit
Character	Data length		Data types and attributes that can be handled (*1)
	Specification length (n)		Up to 10,485,760 characters (*1)
Numeric	External decimal expression		Up to 131,072 digits before the decimal point, and up to 16,383 digits after the decimal point
	Internal binary expression	2 bytes	From -32,768 to 32,767
		4 bytes	From -2,147,483,648 to 2,147,483,647
		8 bytes	From -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
	Internal decimal expression		Up to 13,1072 digits before the decimal point, and up to 16,383 digits after the decimal point
	Floating point expression	4 bytes	From -3.4E+38 to -7.1E-46, 0, or from 7.1E-46 to 3.4E+38
		8 bytes	From -1.7E+308 to -2.5E-324, 0, or from 2.5E-324 to 1.7E+308
bytea			Up to one gigabyte minus 53 bytes
Large object			Up to 4 terabyte

*1: This is the character string byte length when converted by the server character set character code.

Table G.6 Function definition

Item	Limit
Number of arguments that can be specified	Up to 100
Number of variable names that can be specified in the declarations section	No limit
Number of SQL statements or control statements that can be specified in a function processing implementation	No limit

Table G.7 Data operation statement

Item	Limit
Maximum number of connections for one process in an application (remote access)	4,000 connections
Number of expressions that can be specified in a selection list	Up to 1,664
Number of tables that can be specified in a FROM clause	No limit
Number of unique expressions that can be specified in a selection list/ DISTINCT clause/ORDER BY clause/GROUP BY clause within one SELECT statement	Up to 1,664
Number of expressions that can be specified in a GROUP BY clause	No limit
Number of expressions that can be specified in an ORDER BY clause	No limit
Number of SELECT statements that can be specified in a UNION clause/INTERSECT clause/EXCEPT clause	Up to 4,000 (*1)
Number of nestings in joined tables that can be specified in one view	Up to 4,000 (*1)
Number of functions or operator expressions that can be specified in one expression	Up to 4,000 (*1)
Number of expressions that can be specified in one row constructor	Up to 1,664
Number of expressions that can be specified in an UPDATE statement SET clause	Up to 1,664
Number of expressions that can be specified in one row of a VALUES list	Up to 1,664
Number of expressions that can be specified in a RETURNING clause	Up to 1,664
Total expression length that can be specified in the argument list of one function specification	Up to 800 megabytes (*2)
Number of cursors that can be processed simultaneously by one session	No limit
Character string length of one SQL statement	Up to 800 megabytes (*1) (*3)
Number of input parameter specifications that can be specified in one dynamic SQL statement	No limit
Number of tokens that can be specified in one SQL statement	Up to 10,000
Number of values that can be specified as a list in a WHERE clause IN syntax	No limit
Number of expressions that can be specified in a USING clause	No limit
Number of JOINs that can be specified in a joined table	Up to 4,000 (*1)
Number of expressions that can be specified in COALESCE	No limit
Number of WHEN clauses that can be specified for CASE in a simple format or a searched format	No limit
Data size per record that can be updated or inserted by one SQL statement	Up to one gigabyte minus 53 bytes
Number of objects that can share a lock simultaneously	Up to 256,000 (*1)

*1: Operation might proceed correctly even if operations are performed with a quantity outside the limits.

*2: The total number of all database objects must be less than 4,294,967,296.

*3: This is the character string byte length when converted by the server character set character code.

Table G.8 Data size

Item	Limit
Data size per record for input data files (COPY statement, psql command \copy meta command)	Up to 800 megabytes (*1)
Data size per record for output data files (COPY statement, psql command \copy meta command)	Up to 800 megabytes (*1)

*1: Operation might proceed correctly even if operations are performed with a quantity outside the limits.

Appendix H Determining the Preferred WebAdmin Configuration

This appendix describes the two different configurations in which WebAdmin can be used and how to select the most suitable configuration.

H.1 WebAdmin Configurations

WebAdmin can be installed in two configurations:

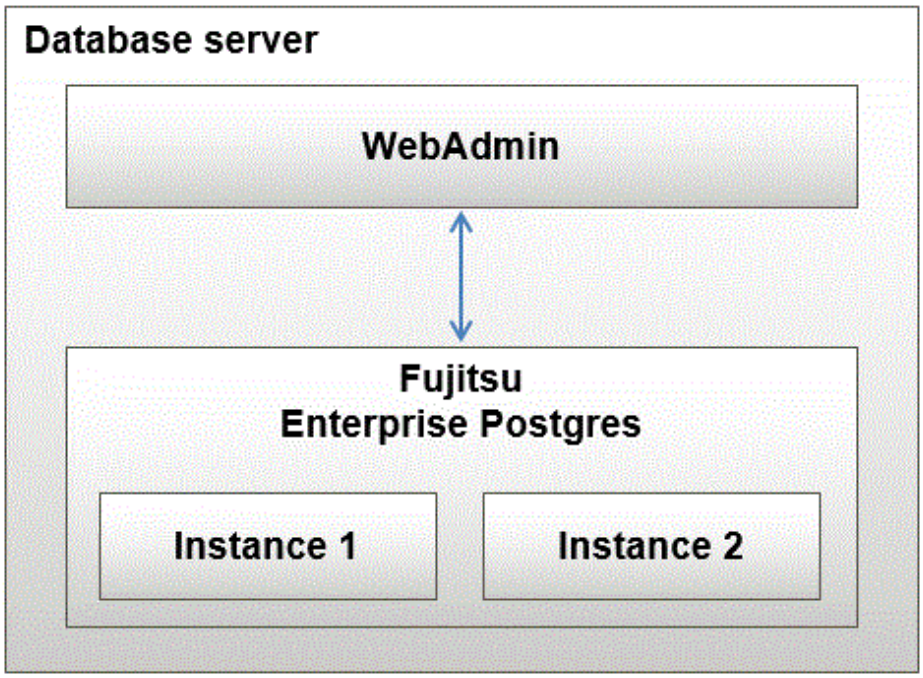
- Single-server
- Multiserver

WebAdmin does not support encrypted communication between browser and server or between servers. Therefore, when using WebAdmin in either configuration, build the communication path with the browser or each server on a network that cannot be accessed externally.

H.1.1 Single-Server Configuration

A single-server configuration enables you to create and operate instances on a single server. In this configuration, WebAdmin must be installed on the same database server as the Fujitsu Enterprise Postgres Server component.

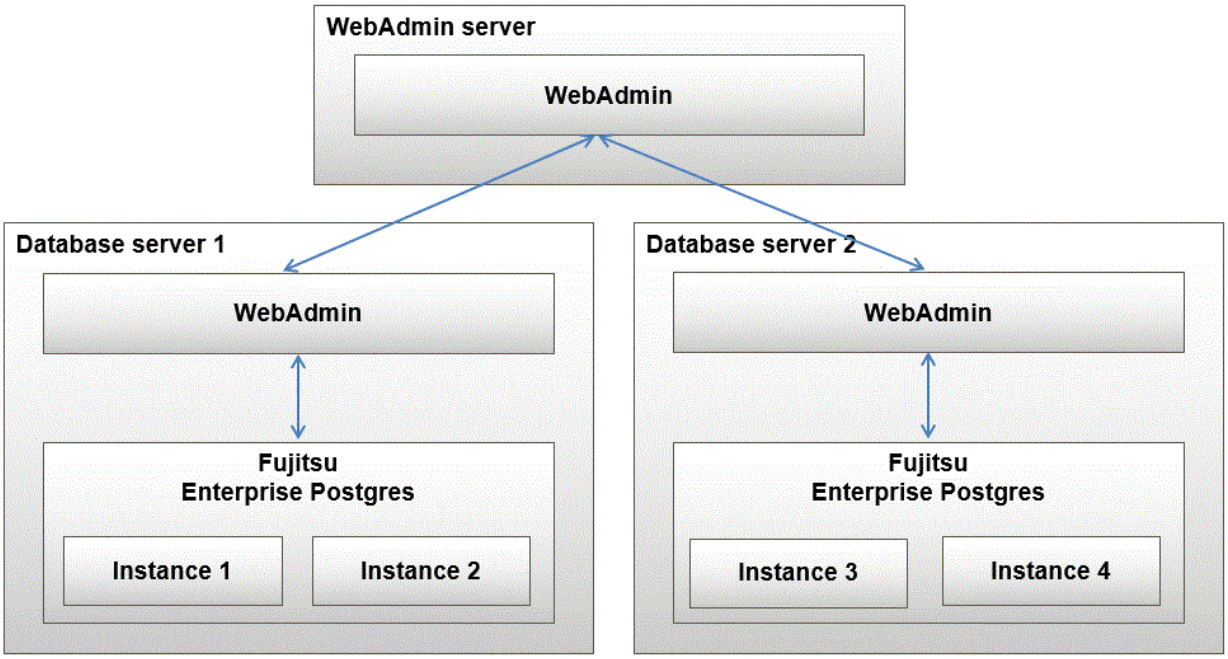
Single-server configuration



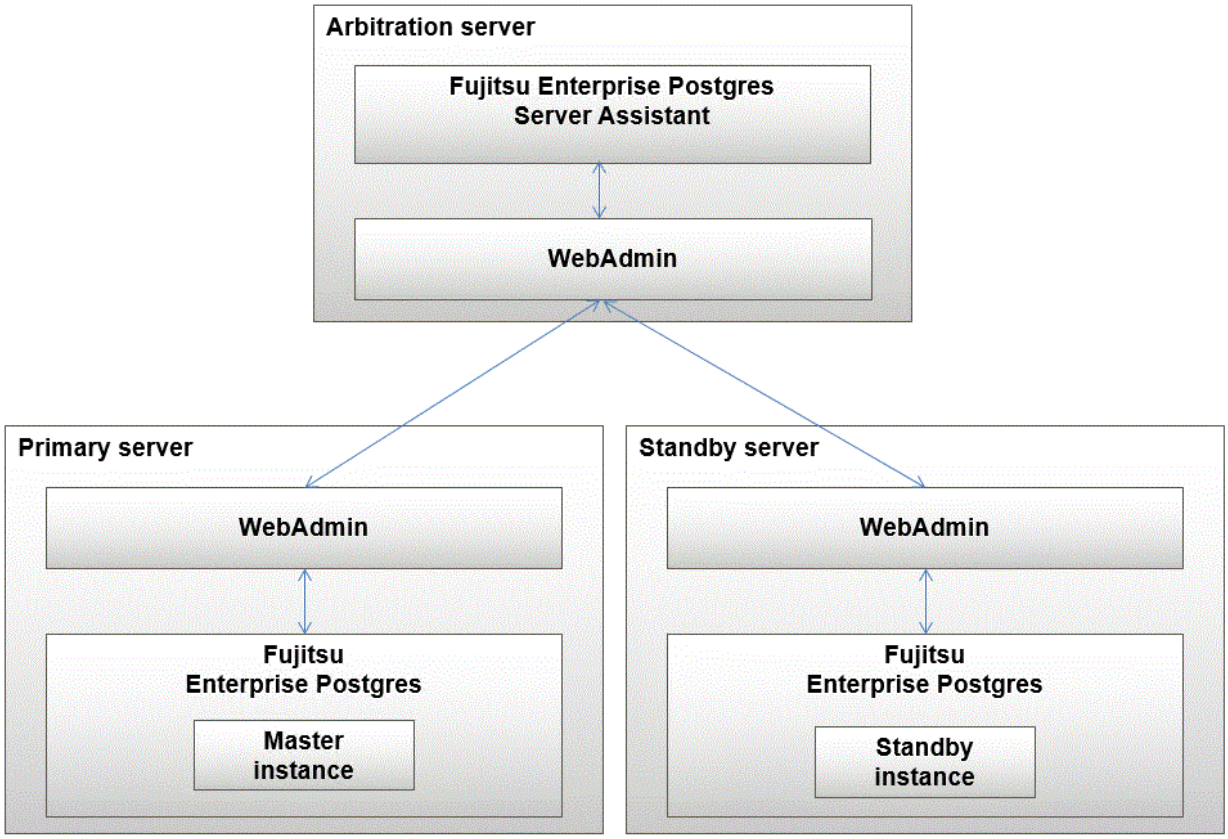
H.1.2 Multiserver Configuration

A multiserver configuration enables you to create and operate instances stored on multiple database servers. As shown in the figure below, WebAdmin can be installed on a dedicated WebAdmin server and used to collectively manage the instances stored on the database servers.

Multiserver configuration



Also, when setting up the arbitration server by WebAdmin during database multiplexing mode, install WebAdmin on the arbitration server.



H.2 Installing WebAdmin in a Single-Server Configuration

To install WebAdmin in a single-server configuration, the Fujitsu Enterprise Postgres Server component and WebAdmin must be installed on the same machine.

Select the following items when installing Fujitsu Enterprise Postgres in a single-server configuration:

- Fujitsu Enterprise Postgres Advanced Edition or Fujitsu Enterprise Postgres Standard Edition
- WebAdmin

H.3 Installing WebAdmin in a Multiserver Configuration

In a multiserver configuration, install WebAdmin on one server, and both WebAdmin and the Fujitsu Enterprise Postgres Server component on any number of database servers.

Select the following items when installing Fujitsu Enterprise Postgres in a multiserver configuration:

- WebAdmin server:
  - WebAdmin
- Database server:
  - Fujitsu Enterprise Postgres Advanced Edition or Fujitsu Enterprise Postgres Standard Edition
  - WebAdmin

Also, when setting up the arbitration server by WebAdmin during database multiplexing mode, select the following when installing Fujitsu Enterprise Postgres.

- Arbitration server
  - Fujitsu Enterprise Postgres Server Assistant
  - WebAdmin

See

Refer to the Installation and Setup Guide for Server Assistant for details on how to install the Server Assistant.

Appendix I Supported contrib Modules and Extensions Provided by External Projects

Fujitsu Enterprise Postgres supports PostgreSQL contrib modules, and extensions provided by external projects.

Refer to the following for details on the supported contrib modules:

- "Additional Supplied Modules" in the PostgreSQL Documentation
- "Additional Supplied Programs" in the PostgreSQL Documentation

Information

You can also check the list of available extensions using the pg_available_extensions view.

Refer to "OSS Supported by Fujitsu Enterprise Postgres" in the General Description for information on supported extensions provided by external projects.

Index

[C]

Changing client authentication information....27
Changing instance settings....26
Check the disk space....8
Check the installed products and determine the installation method....8
Client Authentication Information settings....33
Creating an Instance....23,30
Creating an Instance Administrator....14
Creating Instances....21

[D]

Disk Space Required for Installation....5

[E]

Editing instance information....28
Excluded Software....4

[F]

Firewall....19,33

[H]

Hardware Environment....5
How to Set Up the Pop-up Blocker....48

[I]

Importing Instances....28
Installation....7
Installation in Interactive Mode....2,9
Installation in Silent Mode....2,11
Installation Procedure....1
Installation Types....1
Instance configuration....27

[L]

Logging in to WebAdmin....22

[M]

Multi-Version Installation....1

[N]

New Installation....1

[O]

Operating Environment....3
Operating Method Types and Selection....13

[P]

Port number to use when Tomcat is stopped....49
postgresql.conf....55
Pre-installation Tasks....8
Preparations for Setup....14

[R]

Recommended Browser Settings....48
Reinstallation....1
Related Software....3
Remove applied updates....8
Removing WebAdmin....51
Required Operating System....3
Required Patches....5

[S]

Settings related to connection....33
Setting Up and Removing WebAdmin....49
Setting Up WebAdmin....49
Starting the Web Server Feature of WebAdmin....50
Stopping the Web Server Feature of WebAdmin....50
Supported contrib Modules and Extensions Provided by External Projects....77

[T]

TCP/IP Protocol....5

[U]

Uninstallation....2,42
Uninstallation in Interactive Mode....42
Uninstallation in Silent Mode....44
uninstaller....47
Using the initdb Command....30
Using WebAdmin....22

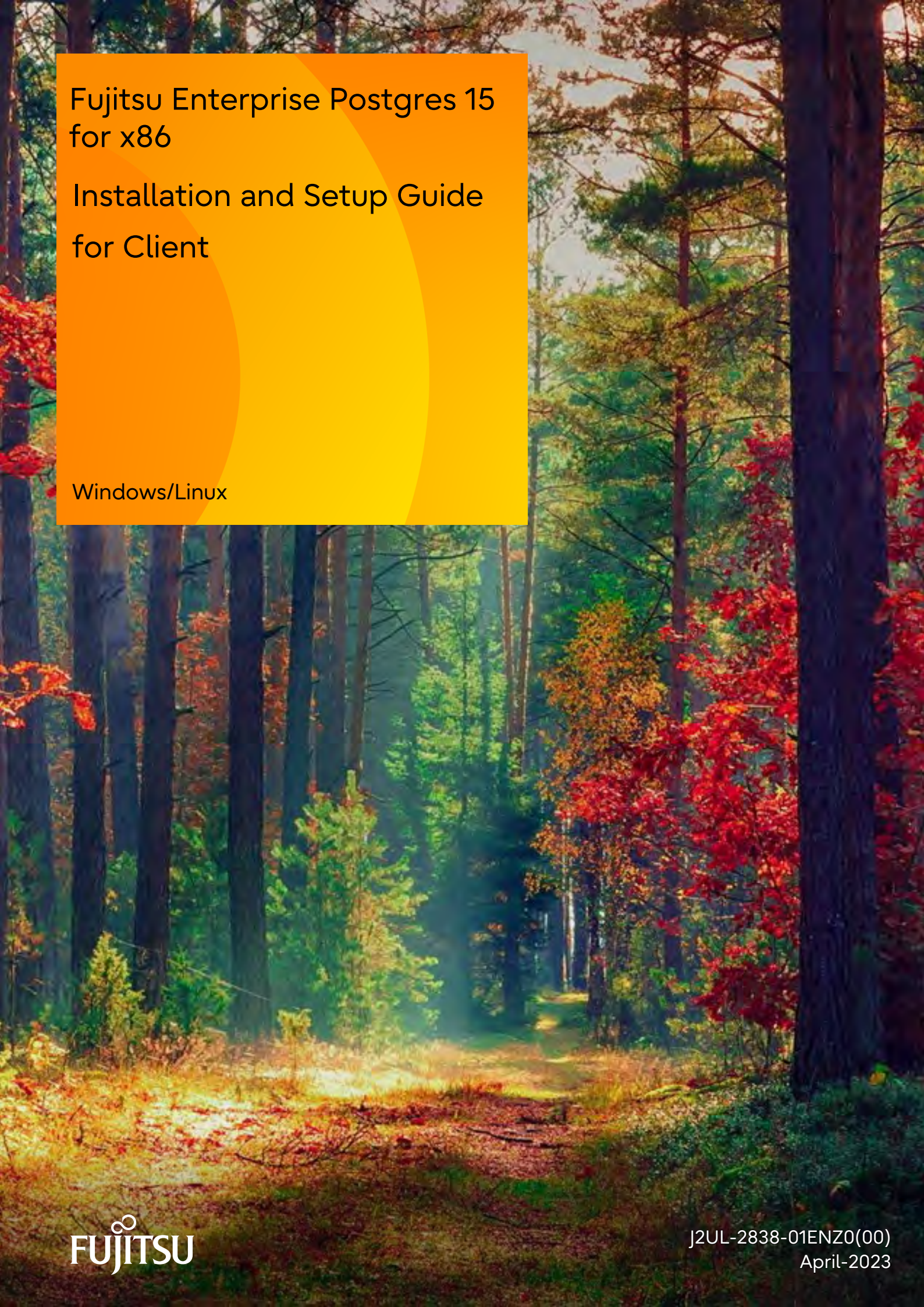
[W]

WebAdmin automatic start....49
Web server port number....49
When an Instance was Created with the initdb Command....33
When an Instance was Created with WebAdmin....33

Fujitsu Enterprise Postgres 15 for x86

Installation and Setup Guide for Client

Windows/Linux



FUJITSU

J2UL-2838-01ENZ0(00)
April-2023

Preface

Purpose of this document

This document describes how to install, uninstall and set up the "Fujitsu Enterprise Postgres client feature".

Intended readers

This document is intended for those who install and operate Fujitsu Enterprise Postgres.

Readers of this document are assumed to have general knowledge of:

W
- PostgreSQL
- SQL
- Windows

L
- PostgreSQL
- SQL
- Linux

Structure of this document

This document is structured as follows:

Chapter 1 Overview of Installation

Describes the features that can be installed, and provides an overview of installation methods

Chapter 2 Installation and Uninstallation of the Windows Client

Describes how to install the Fujitsu Enterprise Postgres client feature (Windows client)

Chapter 3 Installation and Uninstallation of the Linux Client

Describes how to install the Fujitsu Enterprise Postgres client feature (Linux client)

Chapter 4 Setup

Describes the setup procedures to be performed after installation completes

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

Edition 1.0: April 2023

Copyright

Copyright 2015-2023 Fujitsu Limited

Contents

Chapter 1 Overview of Installation....1
1.1 Features that can be Installed....1
1.2 Installation Types....1
1.2.1 New Installation....1
1.2.2 Reinstallation....1
1.2.3 Multi-Version Installation....1
1.3 Installation Procedure....1
1.3.1 Installation in Interactive Mode....2
1.3.2 Installation in Silent Mode....2
1.4 Uninstallation....2

Chapter 2 Installation and Uninstallation of the Windows Client....3
2.1 Operating Environment....3
2.1.1 Required Operating System....3
2.1.2 Related Software....3
2.1.3 Excluded Software....5
2.1.4 Required Patches....5
2.1.5 Hardware Environment....5
2.1.6 Disk Space Required for Installation....5
2.1.7 Supported System Environment....5
2.1.8 Versions of Open-Source Software Used as the Base for Fujitsu Enterprise Postgres Drivers....6
2.2 Installation....6
2.2.1 Pre-installation Tasks....6
2.2.2 Pre-installation Considerations....7
2.2.3 Installation in Interactive Mode....9
2.2.4 Installation in Silent Mode....10
2.3 Uninstallation....11
2.3.1 Uninstallation in Interactive Mode....11
2.3.2 Uninstallation in Silent Mode....11

Chapter 3 Installation and Uninstallation of the Linux Client....13
3.1 Operating Environment....13
3.1.1 Required Operating System....13
3.1.2 Related Software....15
3.1.3 Excluded Software....17
3.1.4 Required Patches....17
3.1.5 Hardware Environment....17
3.1.6 Disk Space Required for Installation....17
3.1.7 Supported System Environment....17
3.1.8 Versions of Open-Source Software Used as the Base for Fujitsu Enterprise Postgres Drivers....18
3.2 Installation....18
3.2.1 Pre-installation Tasks....18
3.2.2 Installation in Interactive Mode....20
3.2.3 Installation in Silent Mode....21
3.3 Uninstallation....22
3.3.1 Uninstallation in Interactive Mode....22
3.3.2 Uninstallation in Silent Mode....23

Chapter 4 Setup....25
4.1 Configuring Environment Variables....25
4.1.1 Windows....25
4.1.2 Linux....25

Index....26

Chapter 1 Overview of Installation

This chapter provides an overview of Fujitsu Enterprise Postgres installation.

1.1 Features that can be Installed

Fujitsu Enterprise Postgres provides features to enable access to the database from a variety of platforms and languages, as the connection environment for the client and the database server.

The Fujitsu Enterprise Postgres client package must be installed on the client system to use these features.

The following table shows the relationship between the platforms and the features provided by client packages.

Platform	JDBC	ODBC	.NET Data Provider	C language (libpq)	Embedded SQL (ECPG) in C language	Embedded SQL (ECOBPG) in COBOL	Connection Manager	High-speed data load	pgAdmin	Pgpool-II
Windows	Y	Y	Y	Y	Y	Y	Y	Y	Y	
Linux	Y	Y		Y	Y	Y	Y	Y		Y

Y: Supported

1.2 Installation Types

The following installation types are available for Fujitsu Enterprise Postgres:

- New installation
- Reinstallation
- Multi-version installation

1.2.1 New Installation

In initial installation, the Fujitsu Enterprise Postgres client feature is installed for the first time.

1.2.2 Reinstallation

Perform reinstallation to repair installed program files that have become unusable for any reason.

1.2.3 Multi-Version Installation

Perform multi-version installation to install different versions to the installed program files separately.

1.3 Installation Procedure

The following installation procedures are available for Fujitsu Enterprise Postgres:

- Installation in interactive mode
- Installation in silent mode

Select the installation procedure that corresponds to your environment.

1.3.1 Installation in Interactive Mode

Interactive mode enables installation to be performed while the required information is entered interactively.

In interactive mode installation, the Fujitsu Enterprise Postgres installer automatically determines the installation state of Fujitsu Enterprise Postgres. Install Fujitsu Enterprise Postgres using one of the following installation types in accordance with the installation state:

- New installation
- Reinstallation
- Multi-version installation

1.3.2 Installation in Silent Mode

Silent mode enables installation to be performed without the need to enter any information interactively.

New installations and multi-version installations can be performed in silent mode.

1.4 Uninstallation

Uninstallation removes the system files of the installed Fujitsu Enterprise Postgres client feature.

Chapter 2 Installation and Uninstallation of the Windows Client

This chapter explains how to install and uninstall the Windows client.

2.1 Operating Environment

This section describes the operating environment required to use the Windows client.

2.1.1 Required Operating System

One of the following operating systems is required to use the Windows client:

- Windows(R) 10 Home
- Windows(R) 10 Education
- Windows(R) 10 Pro
- Windows(R) 10 Enterprise
- Windows(R) 11 Home
- Windows(R) 11 Education
- Windows(R) 11 Pro
- Windows(R) 11 Enterprise
- Microsoft Windows Server 2016 Datacenter
- Microsoft Windows Server 2016 Standard
- Microsoft Windows Server 2016 Essentials
- Microsoft Windows Server 2019 Datacenter
- Microsoft Windows Server 2019 Standard
- Microsoft Windows Server 2019 Essentials
- Microsoft Windows Server 2022 Datacenter
- Microsoft Windows Server 2022 Standard
- Microsoft Windows Server 2022 Essentials

Note: If Windows is 32 bit, only the Windows client (32 bit) can be installed.

Note

The following components of Windows Server(R) 2016, Windows Server(R) 2019 and Windows Server(R) 2022 are not supported:

- Server Core
- Nano Server
- Windows Server Container

2.1.2 Related Software

The following table lists the software compatible (that can operate) with the Windows client.

Before using any of these, confirm that the OS supports the software.

Table 2.1 Related software

No.	Software name	Version
1	Visual Studio	2015 2017 2019 2022
2	.NET Framework	.NET 6.0 .NET 5.0 .NET Core 3.1 .NET Standard 2.0 .NET Standard 2.1
3	C compiler (*1)	-
4	JDK or JRE	Java SE 8 or later
5	NetCOBOL NetCOBOL for .NET	(*2) (*3)

*1: Only operations using the C compiler provided with the operating system are guaranteed.

*2: NetCOBOL is available in the following editions:

- NetCOBOL Base Edition V10.5.0 or later
- NetCOBOL Standard Edition V10.5.0 or later
- NetCOBOL Professional Edition V10.5.0 or later
- NetCOBOL Enterprise Edition V10.5.0 or later

*3: NetCOBOL for .NET is available in the following editions:

- NetCOBOL Standard Edition for .NET V5.0.0 or later
- NetCOBOL Professional Edition for .NET V5.0.0 or later
- NetCOBOL Enterprise Edition for .NET V5.0.0 or later

Note

The following JDKs or JREs are available:

- JDK or JRE shipped with the Interstage Application Server
- JDK shipped with the Enterprise Application Platform
- Oracle JDK or JRE
- An OpenJDK that has passed the Java TCK (Technology Compatibility Kit)

It is recommended that the JDK or JRE included with Interstage Application Server and Enterprise Application Platform is used.

The following table lists servers that can be connected to the Windows client.

Table 2.2 Connectable servers

OS	Software name
Windows Linux	- FUJITSU Software Enterprise Postgres Advanced Edition 9.5 or later , up to 15

OS	Software name
	- FUJITSU Software Enterprise Postgres Standard Edition 9.4 or later , up to 15
Solaris	- FUJITSU Software Enterprise Postgres Standard Edition 9.6

Note

Connecting this client to a server product of a different version depends on compatibility with PostgreSQL on which the server product is based, so some features may not be available.

2.1.3 Excluded Software

Fujitsu Enterprise Postgres

The Windows client cannot coexist with the Fujitsu Enterprise Postgres Community Edition (hereafter, "CE") client.

Other products

There are no exclusive products.

2.1.4 Required Patches

No patches are required.

2.1.5 Hardware Environment

The following hardware is required to use the Windows client.

Memory

At least 256 MB of memory is required.

Mandatory hardware

None.

2.1.6 Disk Space Required for Installation

The disk space shown below is required for new installation of the Windows client.

At the Windows (32-bit) client installation destination:

180 MB

At the Windows (64-bit) client installation destination:

980 MB

System folder:

256 MB

2.1.7 Supported System Environment

This section describes the supported system environment.

TCP/IP protocol

Fujitsu Enterprise Postgres supports version 4 and 6 (IPv4 and IPv6) of TCP/IP protocols.

Note

Do not use link-local addresses if TCP/IP protocol version 6 addresses are used.

File system

You can install Fujitsu Enterprise Postgres only if the system folder is an NTFS volume.

2.1.8 Versions of Open-Source Software Used as the Base for Fujitsu Enterprise Postgres Drivers

The following table lists the versions of open-source software used as the base for the various Fujitsu Enterprise Postgres drivers.

Driver	Open-source software version
.NET Data Provider	Npgsql 6.0.7
JDBC	pgjdbc 42.5.0
ODBC	psqlodbc 13.02.0000
libpq	PostgreSQL 15.0

2.2 Installation

This section explains how to install the Windows client.

Note

When installing the Fujitsu Enterprise Postgres Client (32 bit) in a 64 bit environment, do not specify a destination folder under the environment ProgramFiles variable.

2.2.1 Pre-installation Tasks

This section describes the operations to be performed before installing the Windows client.

hosts file settings

Using an editor, add the IP address and server name of the connected server to the \System32\drivers\etc\hosts file in the folder where Windows is installed.

Note

For Windows(R) 10, or Windows(R) 11, assign write privileges in advance to the user responsible for editing the hosts file.

Check the installed product and determine the installation method

In Windows, click [All Programs] or [All apps], then [Fujitsu], and then [Uninstall (middleware)]. In the displayed window, check the installed products.

If the Windows client is already installed, determine the installation method to use:

- Reinstallation
- Multi-version installation

Remove applied updates

If you perform reinstallation as the installation method, remove applied updates using the procedure shown below.

Note

If a product is installed without removing applied updates, the following problems will occur:

- Performing reinstallation

  If an update with the same update and version number is applied, an error informing you that the update has already been applied is displayed.

1. Display the applied updates

   Execute the following command to display the applied updates:

```
C:\Program Files (x86)\Fujitsu\UpdateAdvisor\UpdateAdvisorMW>uam showup
```

2. Remove the updates

   Execute the command below to remove the updates. If an update with the same update number was applied more than once, the updates are removed in order, starting from the highest version number.

```
C:\Program Files (x86)\Fujitsu\UpdateAdvisor\UpdateAdvisorMW>uam remove -i update-number
```

Confirm service

The Windows Installer service must be running.

Confirm Path system environment variable

Ensure that the Path system environment variable includes "%SystemRoot%\system32".

2.2.2 Pre-installation Considerations

This section describes points that the user must take into account prior to installation.

- The remote desktop service is installed in application server mode, it is necessary to switch to install mode by executing the command shown below before installation. Also, after the installation is completed, execute the command shown below to switch back to execute mode.

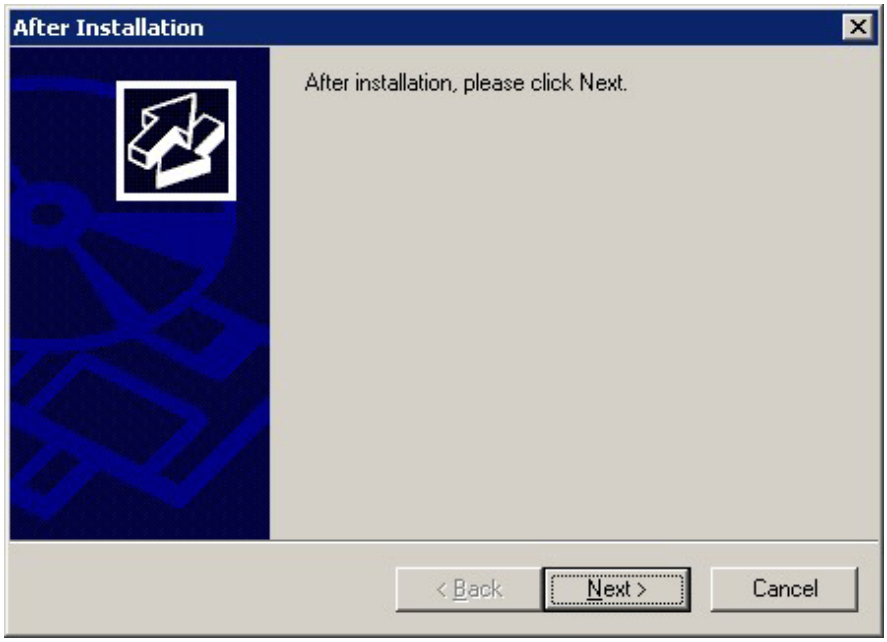
  Before installation:

```
CHANGE USER /INSTALL
```

  After installation:

```
CHANGE USER /EXECUTE
```

- The following window may be displayed when the installation program is executed.



If the above window is displayed, follow the procedure below:

1. Perform the steps in the installation procedure until the [InstallShield Wizard Complete] window is displayed.
2. Click [Next] in the [InstallShield Wizard Complete] window.
3. The following window is displayed. Click [Finish].

4. Restart the system.

Information

- If the [User Account Control] dialog box is displayed when installation or uninstallation starts, click [Yes] to continue the operation.

  If [No] is clicked, the [Error] dialog box will be displayed because continuing is not permitted.
  Click [Retry] in the displayed [Error] dialog box to proceed with installation or uninstallation. To end operations, click [Cancel].

- If installation or uninstallation is suspended or processing terminates abnormally, the [Program Compatibility Assistant] dialog box may be displayed.
  If this happens, click [This program installed correctly] or [This program uninstalled correctly] and continue operation.

2.2.3 Installation in Interactive Mode

The installation procedure is described below.

Point

For installation in interactive mode, default values are set for the installation information. The following settings can be changed for a new installation or a multi-version installation:

- Installation folder

1. Stop applications and programs

If the installation method is the following, all applications and programs that use the product must be stopped:

- Reinstallation

Before starting the installation, stop the following:

- Applications that use the product
- Connection Manager
- pgAdmin

2. Insert the DVD

Insert the client program DVD into the DVD drive.

3. Run the installation

The installation menu is displayed. Click [Installation].

4. Select the product to install

Select the product to install, and then click [Next].
If a selected product can only be reinstalled, refer to "6. Check the settings".

Information

If multi-version installation is possible, the [Select Installation Method] window is displayed for each selected product. Select "Multi-version installation" and click [Next].

5. Confirm the contents for installation

The [Confirm installation] window will be displayed.

Click [Next] to start the installation.

To modify the settings, select [Modify], and then click [Next]. Follow the on-screen instructions.

6. Check the settings

The [Confirm installation] window will be displayed for reinstallation, or if the installation information is modified.

Click [Install] to start the installation.

To modify the settings again, click [Back].

7. Completion of installation

The completion window is displayed. Click [Finish].

2.2.4 Installation in Silent Mode

Installation in silent mode can be performed only when the installation method is one of the following:

- New installation
- Multi-version installation

See

Refer to the Fujitsu Enterprise Postgres product website for information on installation in silent mode, such as the installation parameters and error messages.

The installation procedure is described below.

1. Insert the DVD

Insert the client program DVD into the DVD drive.

The [Install Menu] window is displayed. Click [Finish].

2. Create an installation parameters CSV file

Consider the server type or features that will be required for system operations, and then create an installation parameters CSV file that uses the following specification format.

```
sectionName, parameterName, value
sectionName, parameterName, value
        :
```

Information

The template for the installation parameters CSV file is "*Z*:\sample\sample_windows.csv (*Z* is the drive into which the DVD is inserted.).

3. Start the command prompt

In Windows, right-click [Command Prompt] and then select [Run as administrator].

4. Run the installation

Execute the following command:

```
Z:\>silent.bat c:\temp\inspara.csv
```

Z: The drive into which the DVD is inserted.

c:\temp\inspara.csv: The installation parameter CSV file name.

If the silent installer ends in an error, a message is output to the log file and return values are returned.

2.3 Uninstallation

This section describes the procedure for uninstalling the Windows client.

Note

- Before uninstalling the product, close the product program and all applications that are using it.
- Log in using an account that has administrator privileges and then execute the command, or switch to an account that has administrator privileges and then uninstall the product.

2.3.1 Uninstallation in Interactive Mode

The uninstallation procedure is described below.

Information

If an error occurs while the product is being uninstalled, refer to "Uninstall (middleware) Messages" in the Fujitsu Enterprise Postgres product website, and take the required action.

1. Stop applications and programs

Before starting the uninstallation, stop the following:

- Applications that use the product
- Connection Manager
- pgAdmin

2. Start the Uninstall (middleware) tool

In Windows, click [All Programs] or [All apps], then [Fujitsu], and then [Uninstall (middleware)].

3. Select the software

Select the product to be uninstalled from [Software Name], and then click [Remove].

4. Start the uninstallation

Click [Uninstall].

5. Finish the uninstallation

The uninstallation completion window is displayed. Click [Finish].

The installation folder may remain after uninstallation. If it is not required, delete it.

6. Stop the Uninstall (middleware) tool

The [Uninstall (middleware)] window is displayed. Click [Close].

2.3.2 Uninstallation in Silent Mode

The uninstallation procedure is described below.

See

Refer to the Fujitsu Enterprise Postgres product website for information on uninstallation in silent mode, such as the error messages.

1. Stop applications and programs

Before starting the uninstallation, stop the following:

- Applications that use the product
- Connection Manager
- pgAdmin

2. Start the command prompt

In Windows, right-click [Command Prompt] and then select [Run as administrator].

3. Start the uninstaller

Execute the command below.

The installation folder may remain after uninstallation. If it is not required, delete it.

Example

```
X:> installFolder\suninst.bat
```

X: Drive on which the product is installed

Chapter 3 Installation and Uninstallation of the Linux Client

This chapter explains how to install and uninstall the Linux client.

3.1 Operating Environment

This section describes the operating environment required to use the Linux client.

3.1.1 Required Operating System

One of the following operating systems is required to use the Linux client:

- RHEL7.4 or later minor version
- RHEL8.2 or later minor version
- RHEL9.0 or later minor version
- SLES 12 SP5
- SLES 15 SP3 or later minor version

Information

- The following packages are required for operations on RHEL7.

Package name	Remarks
glibc	-
glibc.i686	-
libgcc	-
libmemcached	Required when using Pgpool-II.
libstdc++	-
libtool-ltdl	-
libzstd	-
ncurses-libs	-
nss-softokn-freebl	-
redhat-lsb	-
rsync	Required when using Pgpool-II.
unixODBC	Required when using ODBC drivers
unzip	-
xz-libs	-
zlib	-

- The following packages are required for operations on RHEL8.

Package name	Remarks
glibc	-
glibc.i686	-
libnsl2	-

Package name	Remarks
libgcc	-
libmemcached	Required when using Pgpool-II.
libstdc++	-
libtool-ltdl	-
libzstd	-
ncurses-libs	-
nss-softokn-freebl	-
redhat-lsb	-
rsync	Required when using Pgpool-II.
unixODBC	Required when using ODBC drivers
unzip	-
xz-libs	-
zlib	-

- The following packages are required for operations on RHEL9.

Package name	Remarks
glibc	-
glibc.i686	-
libnsl2	-
libgcc	-
libmemcached-awesome	Required if Pgpool-II is used.
libstdc++	-
libtool-ltdl	-
libzstd	-
ncurses-libs	-
nss-softokn-freebl	-
rsync	Required if Pgpool-II is used.
unixODBC	Required if you are using an ODBC driver.
Unzip	-
xz-libs	-
zlib	-

- The following packages are required for operations on SLES 12.

Package name	Remarks
glibc	-
glibc-32bit	-
JRE 8	Required by installer The following JREs are available: - Oracle JRE Use update 31 or later of the 64-bit version.

Package name	Remarks
	- OpenJDK Past the Java TCK (Technology Compatibility Kit)
libfreebl3	-
libgcc	-
libltdl7	-
liblzma5	-
libmemcached	Required when using Pgpool-II.
libncurses5	-
libstdc++	-
libz1	-
libzstd1	-
rsync	Required when using Pgpool-II.
unixODBC	Required when using ODBC drivers
unzip	-

- The following packages are required for operations on SLES 15.

Package name	Remarks
glibc	-
glibc-32bit	-
JRE 8	Required by installer The following JREs are available: - Oracle JRE Use update 31 or later of the 64-bit version. - OpenJDK Past the Java TCK (Technology Compatibility Kit)
libfreebl3	-
libgcc	-
libltdl7	-
liblzma5	-
libmemcached	Required when using Pgpool-II.
libncurses5	-
libstdc++	-
libz1	-
libzstd1	-
rsync	Required when using Pgpool-II.
unixODBC	Required when using ODBC drivers
unzip	-

3.1.2 Related Software

The following table lists the software required to use the Linux client.

Table 3.1 Related software

No.	Software name	Version
1	C compiler (*1)	-
2	JDK or JRE	Java SE 8 or later
3	NetCOBOL	(*2)

*1: Only operations using the C compiler provided with the operating system are guaranteed.

*2: NetCOBOL is available in the following editions:

(RHEL8)

- NetCOBOL Base Edition V12.2.0 or later
- NetCOBOL Standard Edition V12.2.0 or later
- NetCOBOL Enterprise Edition V12.2.0 or later

(RHEL7)

- NetCOBOL Base Edition V11.1.0 or later
- NetCOBOL Standard Edition V11.1.0 or later
- NetCOBOL Enterprise Edition V11.1.0 or later

Note

The following JDKs or JREs are available:

- JDK or JRE shipped with the Interstage Application Server
- JDK shipped with the Enterprise Application Platform
- Oracle JDK or JRE
- An OpenJDK that has passed the Java TCK (Technology Compatibility Kit)

It is recommended that the JDK or JRE included with Interstage Application Server and Enterprise Application Platform is used.

The following table lists servers that can be connected to the Linux client.

Table 3.2 Connectable servers

OS	Software name
Windows Linux	- FUJITSU Software Enterprise Postgres Advanced Edition 9.5 or later , up to 15 - FUJITSU Software Enterprise Postgres Standard Edition 9.4 or later , up to 15
Solaris	- FUJITSU Software Enterprise Postgres Standard Edition 9.6

Note

Connecting this client to a server product of a different version depends on compatibility with PostgreSQL on which the server product is based, so some features may not be available.

3.1.3 Excluded Software

Fujitsu Enterprise Postgres

The Linux client cannot coexist with the Fujitsu Enterprise Postgres Community Edition (hereafter, "CE") client.

Other products

There are no exclusive products.

3.1.4 Required Patches

There are no required patches.

3.1.5 Hardware Environment

The following hardware is required to use the Linux client.

Memory

At least 160 MB of memory is required.

Mandatory hardware

None.

3.1.6 Disk Space Required for Installation

The following table lists the disk space requirements of the corresponding directories for new installation of the Linux client. If necessary, increase the size of the file system.

RHEL

Directory	Required disk space Unit: MB
/etc	1 + 1 (*1)
/var	1 + 2 (*1)
/opt	283 (*1)
Installation destination of the client (64-bit)	160
Installation destination of Pgpool-II	35

*1: Uninstall (middleware) must be installed.

SLES

Directory	Required disk space Unit: MB
/etc	1 + 1 (*1)
/var	1 + 2 (*1)
/opt	2 (*1)
Installation destination of the client (64-bit)	150
Installation destination of Pgpool-II	35

*1: Uninstall (middleware) must be installed.

3.1.7 Supported System Environment

This section describes the supported system environment.

TCP/IP protocol

Fujitsu Enterprise Postgres supports version 4 and 6 (IPv4 and IPv6) of TCP/IP protocols.

Note

Do not use link-local addresses if TCP/IP protocol version 6 addresses are used.

3.1.8 Versions of Open-Source Software Used as the Base for Fujitsu Enterprise Postgres Drivers

The following table lists the versions of open-source software used as the base for the various Fujitsu Enterprise Postgres drivers.

Driver	Open-source software version
JDBC	pgjdbc 42.5.0
ODBC	psqlodbc 13.02.0000
libpq	PostgreSQL 15.0

3.2 Installation

This section explains how to install the Linux client.

3.2.1 Pre-installation Tasks

Check the system environment for the following before the Linux client is installed.

Check the disk capacity

Check if sufficient free disk space is available for installing the Linux client.

Refer to "3.1.6 Disk Space Required for Installation" for information on disk space requirements.

If sufficient free disk space is unavailable, reconfigure disk partitions.

Set JAVA_HOME (SLES only)

Ensure that JRE 8 is installed, and export the JAVA_HOME environment variable.

```
#export JAVA_HOME="Jre8InstallDir"
```

Executable Users

Installation and uninstallation is performed by one of the following users.

- superuser

  On the system, run the following command to become superuser.

  ```
  $ su -
  Password:******
  ```

- System Administrator

  Check the OS documentation for system administrators.

  Note

  When you execute a command as a system administrator, there are multiple commands that require superuser privileges.

Check the OS documentation before setting. Specify all in the command field.

Check the installed product and determine the installation method

Using the operation shown below, start Uninstall (middleware), and check the installed products.

Example

```
# /opt/FJSVcir/cimanager.sh -c
Loading Uninstaller...


Currently installed products
1. productName
2. productName

Type [number] to select the software you want to uninstall.
[number,q]
=>q



Exiting Uninstaller.
```

If the Linux client is already installed, determine the installation method to use:

- Reinstallation
- Multi-version installation

Remove applied updates

If you perform reinstallation as the installation method, remove applied updates using the procedure shown below.

Note

If a product is installed without removing applied updates, the following will occur:

- Performing reinstallation

  If an update with the same update and version number is applied, an error informing you that the update has already been applied is displayed.

  Perform the reinstallation after removing the update.

1. Display the applied updates

   Execute the following command to display the applied updates:

   ```
   # /opt/FJSVfupde/bin/uam showup
   ```

2. Remove the updates

   Execute the command below to remove the updates. If an update with the same update number was applied more than once, the updates are removed in order, starting from the highest version number.

   ```
   # /opt/FJSVfupde/bin/uam remove -i update-number
   ```

   Note

   If the installation directory/lib is set in the environment variable LD_LIBRARY_PATH for the running user, remove the installation directory/lib from LD_LIBRARY_PATH.

3.2.2 Installation in Interactive Mode

The installation procedure is described below.

Note

The following characters can be used as input values:

Alphanumeric characters, hyphens, commas and forward slashes

Point

For installation in interactive mode, default values are set for the installation information. The following settings can be changed for a new installation or a multi-version installation:

- Installation directory

1. Stop applications and programs

If the installation method is the following, all applications and programs that use the product must be stopped:

- Reinstallation

Before starting the installation, stop the following:

- Applications that use the product
- Connection Manager
- pgBadger
- Pgpool-II

2. Mount the DVD drive

Insert the client program DVD into the DVD drive, and then execute the following command:

Example

```
# mount -t iso9660 -r -o loop /dev/dvd /media/dvd
```

Here /dev/dvd is the device name for the DVD drive (which may vary depending on your environment), and /media/dvd is the mount point (which may need to be created before calling the command).

Note

If the DVD was mounted automatically using the automatic mount daemon (autofs), "noexec" is set as the mount option, so the installer may fail to start. In this case, use the mount command to remount the DVD correctly, and then run the installation. Note that the mount options of a mounted DVD can be checked by executing the mount command without any arguments.

3. Run the installation

Execute the following command:

Example

```
# cd /media/dvd
# ./install.sh
```

In the example above, /media/dvd is the DVD mount point.

4. Select the product to install

The list of installation target products is displayed.
Type the number for the product to be installed, or "all", and press Enter.

Information

If the selected product has already been installed, a window for selecting reinstallation or multi-version installation is displayed for each product. Follow the on-screen instructions to select the installation method.

5. Check the settings

The window for checking the installation information is displayed as shown below.
Type "y" and press Enter to start the installation.
To change the settings, type "c" and press Enter, and follow the on-screen instructions. This option is not displayed if there is no information that can be modified.

6. Check the changed settings

If the installation information has been modified, the new installation information is displayed.
Type "y" and press Enter to start the installation.
To change the settings again, type "c" and press Enter.

7. Finish the installation

When the installation is complete, a message is displayed showing the status.
If installation was successful, the following message is displayed:

```
Installed successfully.
```

Note

If an error occurs during the installation, read the error message and remove the cause of the error, and then reexecute the install.sh command.

3.2.3 Installation in Silent Mode

Installation in silent mode can be performed only when the installation method is one of the following:

- New installation
- Multi-version installation

See

Refer to the Fujitsu Enterprise Postgres product website for information on installation in silent mode, such as the installation parameters and error messages.

The installation procedure is described below.

1. Mount the DVD drive

Insert the client program DVD into the DVD drive, and then execute the following command:

Example

```
# mount -t iso9660 -r -o loop /dev/dvd /media/dvd
```

Here /dev/dvd is the device name for the DVD drive (which may vary depending on your environment), and /media/dvd is the mount point (which may need to be created before calling the command).

Note

If the DVD was mounted automatically using the automatic mount daemon (autofs), "noexec" is set as the mount option, so the installer may fail to start. In this case, use the mount command to remount the DVD correctly, and then run the installation. Note that the mount options of a mounted DVD can be checked by executing the mount command without any arguments.

2. Create an installation parameters CSV file

Consider the features that will be required for system operations, and then create an installation parameters CSV file that uses the following specification format.

```
sectionName, parameterName, value
sectionName, parameterName, value
    :
```

Information

The templates for the installation parameters CSV file is "*mountPoint*/sample/sample_linux.csv"

3. Run the installation

Execute the following command:

Example

```
# cd /media/dvd
# ./silent.sh /home/work/inspara.csv
```

In the example above, /media/dvd is the DVD mount point, and /home/work/inspara.csv is the installation parameter CSV.

If the silent installer ends in an error, a message is output to the log file and return values are returned.

3.3 Uninstallation

This section describes the procedure for uninstalling the Linux client.

Note

- Before uninstalling the product, close the product program and all applications that are using it.
- On SLES, before uninstallation, ensure that JRE 8 is installed, and export the JAVA_HOME environment variable.

```
#export JAVA_HOME="Jre8InstallDir"
```

3.3.1 Uninstallation in Interactive Mode

The uninstallation procedure is described below.

Information

If an error occurs while the product is being uninstalled, refer to "Uninstall (middleware) Messages" in the Fujitsu Enterprise Postgres product website, and take the required action.

1. Stop applications and programs

Before starting the uninstallation, stop the following:

- Applications that use the product
- Connection Manager
- pgBadger
- Pgpool-II

2. Start the Uninstall (middleware)

Execute the following command:

```
# /opt/FJSVcir/cimanager.sh -c
```

3. Select the product

Enter the number for the product to be uninstalled, and press Enter.

4. Start the uninstallation

To start the uninstallation, type "y" and press Enter.

To display the list of products again, type "b" and press Enter.

5. Finish the uninstallation

If uninstallation is successful, the message below is displayed.

The installation directory may remain after uninstallation. If it is not required, delete it.

```
Uninstalling...

productName is being uninstalled now.
    100% ##############################


The following products have been uninstalled successfully:
    productName

Uninstallation of "productName" has completed successfully.



Exiting Uninstaller.
```

3.3.2 Uninstallation in Silent Mode

The uninstallation procedure is described below.

See

Refer to the Fujitsu Enterprise Postgres product website for information on uninstallation in silent mode, such as the error messages.

1. Stop applications and programs

Before starting the uninstallation, stop the following:

- Applications that use the product

- Connection Manager
- pgBadger
- Pgpool-II

2. Run the uninstallation

Execute the following command:

The installation directory may remain after uninstallation. If it is not required, delete it.

Example

```
# /opt/fsepv<x>client64/setup/suninst.sh
```

In the example above, /opt/fsepv<*x*>client64 is the installation directory, and "<*x*>" indicates the product version.

```
# /opt/fsepv<x>pgpool-II/setup/suninst.sh
```

In the example above, /opt/fsepv<*x*>pgpool-II/setup is the name of the installation directory in which the Pgpool-II is installed.

Chapter 4 Setup

This chapter describes the setup procedures to be performed after installation completes.

4.1 Configuring Environment Variables

Configure the following environment variables when using client commands.

4.1.1 Windows

PATH environment variable

Add "*installationFolder*\bin".

PGLOCALEDIR environment variable

Add "*installationFolder*\share\locale".

Examples of environment variable configurations are shown below.

Example

This example is specific to 32-bit Windows.

Note that "<*x*>" indicates the product version.

```
> SET PATH=%ProgramFiles%\Fujitsu\fsepv<x>client32\bin;%PATH%
> SET PGLOCALEDIR=%ProgramFiles%\Fujitsu\fsepv<x>client32\share\locale
```

4.1.2 Linux

PATH environment variable

Add "*installationDirectory*/bin".

MANPATH environment variable

Add "*installationDirectory*/share/man".

PGLOCALEDIR environment variable

Add "*installationDirectory*/share/locale".

Examples of environment variable configurations are shown below.

Example

Note that "<*x*>" indicates the product version.

```
$ PATH=/opt/fsepv<x>client64/bin:$PATH ; export PATH
$ MANPATH=/opt/fsepv<x>client64/share/man:$MANPATH ; export MANPATH
$ PGLOCALEDIR=/opt/fsepv<x>client64/share/locale ; export PGLOCALEDIR
```

Index

[C]

Check the disk capacity....18
Configuring Environment Variables....25
Confirm service....7
Connectable servers....4

[D]

Disk Space Required for Installation....5,17

[E]

Excluded Software....5,17

[F]

Features that can be Installed....1
File system....6

[H]

Hardware Environment....5,17
hosts file settings....6

[I]

Installation....6
Installation and Uninstallation of the Linux Client....13
Installation and Uninstallation of the Windows Client....3
Installation in Interactive Mode....2,9,11
Installation in Silent Mode....2,21
Installation Procedure....1
Installation Types....1

[M]

MANPATH environment variable....25
Multi-Version Installation....1

[N]

New Installation....1

[O]

Operating Environment....3,13

[P]

PATH environment variable....25
PGLOCALEDIR environment variable....25
Pre-installation Considerations....7
Pre-installation Tasks....18

[R]

Reinstallation....1
Related Software....3,15
Required Operating System....3,13
Required Patches....5,17

[S]

Setup....25
Supported System Environment....17

[T]

TCP/IP protocol....5,18

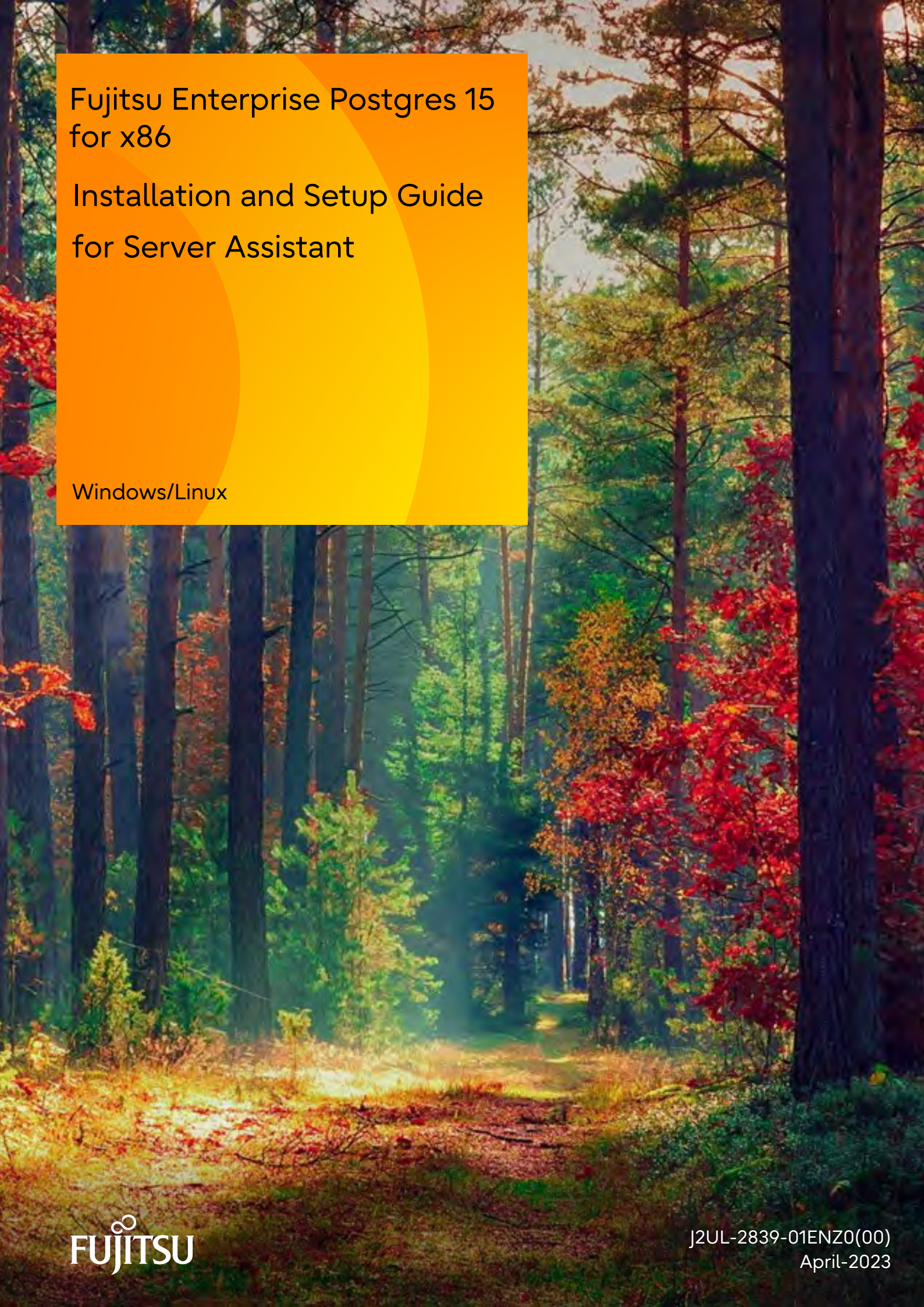
[U]

Uninstallation....2,11,22
Uninstallation in Interactive Mode....22
Uninstallation in Silent Mode....11,23

Fujitsu Enterprise Postgres 15 for x86

Installation and Setup Guide for Server Assistant

Windows/Linux



FUJITSU

J2UL-2839-01ENZ0(00)
April-2023

Preface

Purpose of this document

This document describes how to install and uninstall the Fujitsu Enterprise Postgres Server Assistant.

Intended readers

This document is intended for those who install and operate Fujitsu Enterprise Postgres.

Readers of this document are assumed to have general knowledge of:

L
- PostgreSQL
- SQL
- Linux

W
- PostgreSQL
- SQL
- Windows

Structure of this document

This document is structured as follows:

Chapter 1 Overview of Installation

Describes the features that can be installed, and provides an overview of installation methods

Chapter 2 Installation and Uninstallation of the Linux Server Assistant

Describes how to install and uninstall the Linux Server Assistant

Chapter 3 Installation and Uninstallation of the Windows Server Assistant

Describes how to install and uninstall the Windows Server Assistant

Chapter 4 Setup of the Server Assistant

Describes the setup to be performed after installation

Appendix A Estimating Memory Requirements

Describes the formulas for estimating memory requirements

Appendix B Procedure when Modifying the JRE Installation

Describes the procedure to follow when modifying the JRE installation.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version


```
Edition 1.0: April 2023
```


Copyright

Copyright 2018-2023 Fujitsu Limited

Contents

Chapter 1 Overview of Installation....1
1.1 Features that can be Installed....1
1.2 Installation Types....1
1.2.1 New Installation....1
1.2.2 Reinstallation....1
1.2.3 Multi-Version Installation....1
1.3 Installation Procedure....1
1.3.1 Installation in Interactive Mode....2
1.3.2 Installation in Silent Mode....2
1.4 Uninstallation....2

Chapter 2 Installation and Uninstallation of the Linux Server Assistant....3
2.1 Operating Environment....3
2.1.1 Required Operating System....3
2.1.2 Related Software....5
2.1.3 Excluded Software....6
2.1.4 Required Patches....6
2.1.5 Hardware Environment....6
2.1.6 Disk Space Required for Installation....6
2.1.7 Supported System Environment....6
2.2 Installation....7
2.2.1 Pre-installation Tasks....7
2.2.2 Installation in Interactive Mode....8
2.2.3 Installation in Silent Mode....10
2.3 Uninstallation....11
2.3.1 Uninstallation in Interactive Mode....11
2.3.2 Uninstallation in Silent Mode....12

Chapter 3 Installation and Uninstallation of the Windows Server Assistant....13
3.1 Operating Environment....13
3.1.1 Required Operating System....13
3.1.2 Related Software....13
3.1.3 Excluded Software....14
3.1.4 Required Patches....14
3.1.5 Hardware Environment....14
3.1.6 Disk Space Required for Installation....14
3.1.7 Supported System Environment....14
3.2 Installation....14
3.2.1 Pre-installation Tasks....16
3.2.2 Installation in Interactive Mode....17
3.2.3 Installation in Silent Mode....18
3.3 Uninstallation....19
3.3.1 Uninstallation in Interactive Mode....19
3.3.2 Uninstallation in Silent Mode....21

Chapter 4 Setup of the Server Assistant....23

Appendix A Estimating Memory Requirements....24
A.1 Server Assistant Memory Requirements....24

Appendix B Procedure when Modifying the JRE Installation....25

Index....26

Chapter 1 Overview of Installation

This chapter provides an overview of Fujitsu Enterprise Postgres Server Assistant installation and uninstallation.

1.1 Features that can be Installed

The Server Assistant is provided as a Server Assistant package, which is installed on a different server (referred to as the arbitration server) to that of the database server.

The following table shows the relationship between the platforms and the features provided by the Server Assistant package.

Platform	Server Assistant
Linux	Y
Windows	Y

Y: Can be installed

Note

Use the Linux Server Assistant feature for scale out operations. The Windows Server Assistant feature is not available in a scale out environment.

1.2 Installation Types

The following installation types are available for Fujitsu Enterprise Postgres:

- New installation
- Reinstallation
- Multi-version installation

1.2.1 New Installation

In initial installation, the Fujitsu Enterprise Postgres Server Assistant is installed for the first time.

1.2.2 Reinstallation

Perform reinstallation to repair installed program files that have become unusable for any reason.

1.2.3 Multi-Version Installation

Fujitsu Enterprise Postgres products can be installed on the same server if the product version (indicated by "*x*" in "*x* SP*z*") is different from that of any version of the product that is already installed.

1.3 Installation Procedure

The following installation procedures are available for Fujitsu Enterprise Postgres:

- Installation in interactive mode
- Installation in silent mode

Select the installation procedure that corresponds to your environment.

1.3.1 Installation in Interactive Mode

Interactive mode enables installation to be performed while the required information is entered interactively.

In the interactive mode installation, the Fujitsu Enterprise Postgres installer automatically determines the installation state of Fujitsu Enterprise Postgres. Install Fujitsu Enterprise Postgres using one of the following installation types in accordance with the installation state:

- New installation
- Reinstallation
- Multi-version installation

1.3.2 Installation in Silent Mode

Silent mode enables installation to be performed without the need to enter any information interactively.

New installations and multi-version installations can be performed in silent mode.

1.4 Uninstallation

Uninstallation removes the system files of the installed Fujitsu Enterprise Postgres Server Assistant.

Chapter 2 Installation and Uninstallation of the Linux Server Assistant

This chapter explains how to install and uninstall the Linux Server Assistant.

2.1 Operating Environment

This section describes the operating environment required in order to use the Linux Server Assistant.

2.1.1 Required Operating System

One of the following operating systems is required in order to use the Linux Server Assistant:

- RHEL7.4 or later minor version
- RHEL8.2 or later minor version
- RHEL9.0 or later minor version
- SLES 12 SP5
- SLES 15 SP3 or later minor version

Information

- The following packages are required for operations on RHEL7.

Package name	Remarks
gdb	Required to run FJQSS.
glibc	-
glibc.i686	-
iputils	Required for Mirroring Controller.
libgcc	-
libstdc++	-
ncurses-libs	-
nss-softokn-freebl	-
redhat-lsb	-
rsync	Required when using Pgpool-II.
sysstat	Required when using FJQSS. Set up the sar command after installation.
unzip	-
xz-libs	-
zlib	-

- The following packages are required for operations on RHEL8.

Package name	Remarks
gdb	Required to run FJQSS.
glibc	-
glibc.i686	-

Package name	Remarks
iputils	Required for Mirroring Controller.
libgcc	-
libstdc++	-
ncurses-libs	-
nss-softokn-freebl	-
redhat-lsb	-
rsync	Required when using Pgpool-II.
sysstat	Required when using FJQSS. Set up the sar command after installation.
unzip	-
xz-libs	-
zlib	-

- The following packages are required for operations on RHEL9.

Package name	Remarks
gdb	Required to run FJQSS.
glibc	-
glibc.i686	-
iputils	Required when using Mirroring Controller.
libgcc	-
libstdc++	-
ncurses-libs	-
nss-softokn-freebl	-
rsync	Required to use Pgpool-II.
sysstat	Required if using FJQSS.
unzip	-
xz-libs	-
zlib	-

- The following packages are required for operations on SLES 12.

Package name	Remarks
gdb	Required to run FJQSS.
glibc	-
glibc-32bit	-
iputils	Required for Mirroring Controller.
JRE 8	Required when using the installer or Server Assistant. The following JREs are available: - Oracle JRE Use update 31 or later of the 64-bit version. - OpenJDK Past the Java TCK (Technology Compatibility Kit)
libfreebl3	-

Package name	Remarks
libgcc	-
libltdl7	-
liblzma5	-
libncurses5	-
libstdc++	-
libz1	-
rsync	Required when using Pgpool-II.
sysstat	Required when using FJQSS. Set up the sar command after installation.
unzip	-

- The following packages are required for operations on SLES 15.

Package name	Remarks
gdb	-
glibc	-
glibc-32bit	-
iputils	Required for Mirroring Controller.
JRE 8	Required when using the installer or Server Assistant. The following JREs are available: - Oracle JRE Use update 31 or later of the 64-bit version. - OpenJDK Past the Java TCK (Technology Compatibility Kit)
libfreebl3	-
libgcc	-
libltdl7	-
liblzma5	-
libncurses5	-
libstdc++	-
libz1	-
rsync	Required when using Pgpool-II.
sysstat	Set up the sar command after installation.
unzip	-

2.1.2 Related Software

No other software is required in order to use Fujitsu Enterprise Postgres.

The following table lists servers that can be connected to the Linux Server Assistant.

Table 2.1 Connectable servers

OS	Software name
Linux	Fujitsu Enterprise Postgres Advanced Edition 15
Windows	Fujitsu Enterprise Postgres Advanced Edition 15

2.1.3 Excluded Software

There is no excluded software.

2.1.4 Required Patches

There are no required patches.

2.1.5 Hardware Environment

The following hardware is required in order to use the Linux Server Assistant:

Memory

At least 150 MB of memory is required.

Mandatory hardware

None.

2.1.6 Disk Space Required for Installation

The following table lists the disk space requirements of the corresponding directories for new installation of the Linux Server Assistant. If necessary, increase the size of the file system.

Disk space required for installation

RHEL

Directory	Required disk space Unit: MB
/etc	1 + 1 (*1)
/var	1 + 2 (*1) +1 (*2)
/opt	283 (*1) +1 (*2)
serverAssistantInstallDir	360

*1: Uninstall (middleware) must be installed.

*2: Required for the installation of FJQSS.

SLES

Directory	Required disk space Unit: MB
/etc	1 + 1 (*1)
/var	1 + 2 (*1) +1 (*2)
/opt	2 (*1) +1 (*2)
serverAssistantInstallDir	5

*1: Uninstall (middleware) must be installed.

*2: Required for the installation of FJQSS.

2.1.7 Supported System Environment

This section describes the supported system environment.

TCP/IP Protocol

Fujitsu Enterprise Postgres supports version 4 and 6 (IPv4 and IPv6) of TCP/IP protocols.

Note

Do not use link-local addresses if TCP/IP protocol version 6 addresses are used.

2.2 Installation

This section describes how to install the Linux Server Assistant.

2.2.1 Pre-installation Tasks

Check the following system environment before installing the Linux Server Assistant.

Check the disk space

Ensure that there is sufficient disk space to install the Linux Server Assistant.

Refer to "2.1.6 Disk Space Required for Installation" for information on disk space requirements.

If sufficient free disk space is unavailable, reconfigure disk partitions.

Set JAVA_HOME (SLES only)

Ensure that JRE 8 is installed, and export the JAVA_HOME environment variable.

```
#export JAVA_HOME="Jre8InstallDir"
```

Refer to "Appendix B Procedure when Modifying the JRE Installation" for information on modifying JRE after installation.

Executable Users

Installation and uninstallation is performed by one of the following users.

- superuser

  On the system, run the following command to become superuser.

```
$ su -
Password:******
```

- System Administrator

  Check the OS documentation for system administrators.

  Note

  When you execute a command as a system administrator, there are multiple commands that require superuser privileges.

  Check the OS documentation before setting. Specify all in the command field.

Check the installed product and determine the installation method

Using the operation shown below, start Uninstall (middleware), and check the installed products.

Example

```
# /opt/FJSVcir/cimanager.sh -c
Loading Uninstaller...


Currently installed products
1. productName

Type [number] to select the software you want to uninstall.
```

```
[number,q]
=>q



Exiting Uninstaller.
```

If the Linux Server Assistant is already installed, determine the installation method to use:

- Reinstallation
- Multi-version installation

Remove applied updates

If you perform reinstallation, remove applied updates using the procedure shown below.

Note

If a product is installed without removing applied updates, the following will occur:

- Performing reinstallation

  If an update with the same update and version number is applied, an error informing you that the update has already been applied is displayed.

  Perform reinstallation after removing the update.

1. Display the applied updates

   Execute the following command to display the applied updates:

```
# /opt/FJSVfupde/bin/uam showup
```

2. Remove the updates

   Execute the command below to remove the updates. If an update with the same update number was applied more than once, the updates are removed in order, starting from the highest version number.

```
# /opt/FJSVfupde/bin/uam remove -i update-number
```

   Note

   If the installation directory/lib is set in the environment variable LD_LIBRARY_PATH for the running user, remove the installation directory/lib from LD_LIBRARY_PATH.

2.2.2 Installation in Interactive Mode

The installation procedure is described below.

Note

The following characters can be used as input values:

Alphanumeric characters, hyphens and forward slashes

Point

In interactive mode installation, the default values of the installation information are set. The following setting can be changed for a new installation or a multi-version installation.

- Installation directory

1. Stop the program

If the installation method is the following, the program must be stopped:

- Reinstallation

Before starting the installation, stop the following:

- Mirroring Controller arbitration process

  Execute the mc_arb command in stop mode to stop the Mirroring Controller arbitration process.

  Example

```
$ mc_arb stop -M /mcarb_dir/arbiter1
```

2. Mount the DVD drive

Insert the Server Assistant program DVD into the DVD drive, and then execute the following command:

Example

```
# mount -t iso9660 -r -o loop /dev/dvd /media/dvd
```

Here /dev/dvd is the device name for the DVD drive (which may vary depending on your environment), and /media/dvd is the mount point (which may need to be created before calling the command).

Note

If the DVD was mounted automatically using the automatic mount daemon (autofs), "noexec" is set as the mount option, so the installer may fail to start. In this case, use the mount command to remount the DVD correctly, and then run the installation. Note that the mount options of a mounted DVD can be checked by executing the mount command without any arguments.

3. Run the installation

Execute the following command:

Example

```
# cd /media/dvd
# ./install.sh
```

In the example above, /media/dvd is the DVD mount point.

4. Select the product to install

The list of installation target products is displayed.
Type the number for the product to be installed, or "all", and press Enter.

Information

If the selected product has been installed, a window for selecting reinstallation or multi-version installation is displayed for each product. Follow the on-screen instructions to select the installation method.

5. Check the settings

The window for checking the installation information is displayed.
Type "y" and press Enter to start the installation.

To change the settings, type "c", press Enter, and follow the on-screen instructions. This option is not displayed if there is no information that can be modified.

6. Check the changed settings

If the settings have been changed, the new installation information is displayed.
Type "y" and press Enter to start the installation.
To change the settings again, type "c" and press Enter.

7. Finish the installation

When the installation is complete, a message is displayed showing the status.
If installation was successful, the following message is displayed:

```
Installed successfully.
```

Note

If an error occurs during the installation, read the error message and remove the cause of the error, and then reexecute the install.sh command.

2.2.3 Installation in Silent Mode

Installation in silent mode can be performed only when the installation method is one of the following:

- New installation
- Multi-version installation

See

Refer to the Fujitsu Enterprise Postgres product website for information on installation in silent mode, such as the installation parameters and error messages.

The installation procedure is described below.

1. Mount the DVD drive

Insert the Server Assistant program DVD into the DVD drive, and then execute the following command:

Example

```
# mount -t iso9660 -r -o loop /dev/dvd /media/dvd
```

Here /dev/dvd is the device name for the DVD drive (which may vary depending on your environment), and /media/dvd is the mount point (which may need to be created before calling the command).

Note

If the DVD was mounted automatically using the automatic mount daemon (autofs), "noexec" is set as the mount option, so the installer may fail to start. In this case, use the mount command to remount the DVD correctly, and then run the installation. Note that the mount options of a mounted DVD can be checked by executing the mount command without any arguments.

2. Create an installation parameters CSV file

Create an installation parameters CSV file that uses the following specification format.

```
sectionName, parameterName, value
sectionName, parameterName, value
     :
```

Information

The templates for the installation parameters CSV file is "*mountPoint*/sample/sample.csv"

3. Run the installation

Execute the following command:

Example

```
# cd /media/dvd
# ./silent.sh /home/work/inspara.csv
```

In the example above, /media/dvd is the DVD mount point, and /home/work/inspara.csv is the installation parameter CSV.

If the silent installer ends in an error, a message is output to the log file and return values are returned.

2.3 Uninstallation

This section describes how to uninstall the Linux Server Assistant.

Note

Before uninstalling the product, close the product program.

2.3.1 Uninstallation in Interactive Mode

The uninstallation procedure is described below.

Information

If an error occurs while the product is being uninstalled, refer to "Uninstall (middleware) Messages" in the Fujitsu Enterprise Postgres product website, and take the required action.

1. Stop the program

Before starting the uninstallation, stop the following:

- Mirroring Controller arbitration process

  Execute the mc_arb command in stop mode to stop the Mirroring Controller arbitration process.

  Example

  ```
  $ mc_arb stop -M /mcarb_dir/arbiter1
  ```

2. Start Uninstall (middleware)

Execute the following command:

```
# /opt/FJSVcir/cimanager.sh -c
```

3. Select the product

Enter the number for the product to be uninstalled, and press Enter.

4. Start the uninstallation

To start the uninstallation, type "y" and press Enter.

To display the list of products again, type "b" and press Enter.

5. Finish the uninstallation

If uninstallation is successful, the message below is displayed.

The installation directory may remain after uninstallation. If it is not required, delete it.

```
Uninstalling...

productName is being uninstalled now.
    100% ##############################


The following products have been uninstalled successfully:
    productName

Uninstallation of "productName" has completed successfully.



Exiting Uninstaller.
```

2.3.2 Uninstallation in Silent Mode

The uninstallation procedure is described below.

See

Refer to the Fujitsu Enterprise Postgres product website for information on uninstallation in silent mode, such as the error messages.

1. Stop the program

Before starting the uninstallation, stop the following:

- Mirroring Controller arbitration process

  Execute the mc_arb command in stop mode to stop the Mirroring Controller arbitration process.

  Example

  ```
  $ mc_arb stop -M /mcarb_dir/arbiter1
  ```

2. Run the uninstallation

Execute the following command:

The installation directory may remain after uninstallation. If it is not required, delete it.

Example

```
# /opt/fsepv<x>assistant/setup/suninst.sh
```

In the example above, /opt/fsepv<*x*>assistant is the installation directory, and "<*x*>" indicates the product version.

Chapter 3 Installation and Uninstallation of the Windows Server Assistant

This chapter explains how to install and uninstall the Windows Server Assistant.

3.1 Operating Environment

This section describes the operating environment required in order to use the Windows Server Assistant.

3.1.1 Required Operating System

One of the following operating systems is required in order to use the Windows Server Assistant:

- Microsoft(R) Windows Server(R) 2016 Datacenter
- Microsoft(R) Windows Server(R) 2016 Standard
- Microsoft(R) Windows Server(R) 2016 Essentials
- Microsoft(R) Windows Server(R) 2019 Datacenter
- Microsoft(R) Windows Server(R) 2019 Standard
- Microsoft(R) Windows Server(R) 2019 Essentials
- Microsoft(R) Windows Server(R) 2022 Datacenter
- Microsoft(R) Windows Server(R) 2022 Standard
- Microsoft(R) Windows Server(R) 2022 Essentials

- The following features of Windows Server(R) 2016, Windows Server(R) 2019 and Windows Server(R) 2022 are not supported:
  - Server Core
  - Nano Server
  - Windows Server Container
- The TCP/IP protocol must be installed.

3.1.2 Related Software

There is no software related to Fujitsu Enterprise Postgres.

- The following programs are installed during installation of Fujitsu Enterprise Postgres:
  - Microsoft Visual C++ 2015-2019 Redistributable version 14.24.28127.4

  Do not uninstall the above programs as they are required for running Fujitsu Enterprise Postgres.

The following table lists servers that can be connected to the Windows Server Assistant.

Table 3.1 Connectable servers

OS	Software name
Linux	Fujitsu Enterprise Postgres Advanced Edition 15
Windows	Fujitsu Enterprise Postgres Advanced Edition 15

3.1.3 Excluded Software

There is no excluded software.

3.1.4 Required Patches

There are no required patches.

3.1.5 Hardware Environment

The following hardware is required in order to use the Windows Server Assistant:

Memory

At least 256 MB of memory is required.

Mandatory hardware

None.

3.1.6 Disk Space Required for Installation

The disk space shown below is required for new installation of the Windows Server Assistant.

Windows Server Assistant installation destination:

380 MB

System folder:

285 MB

3.1.7 Supported System Environment

This section describes the supported system environment.

TCP/IP protocol

Fujitsu Enterprise Postgres supports version 4 and 6 (IPv4 and IPv6) of TCP/IP protocols.

Note

Do not use link-local addresses if TCP/IP protocol version 6 addresses are used.

File system

You can install Fujitsu Enterprise Postgres only if the system folder is an NTFS volume.

3.2 Installation

This section describes how to install the Windows Server Assistant.

Note

- The installation must be performed by a user with administrator privileges (a user ID that belongs to the Administrators group).
- Stop all applications before starting the installation.
- The Windows Installer service must be running.
- If antivirus software is installed, the system may crash, fail to start, or stop responding during installation or when starting after installation. Set the scan exclusions so that the installation directory and the resource allocation directory are not scanned for viruses.
- The remote desktop service is installed in application server mode, it is necessary to switch to install mode by executing the command shown below before installation. Also, after the installation is completed, execute the command shown below to switch back to execute mode.

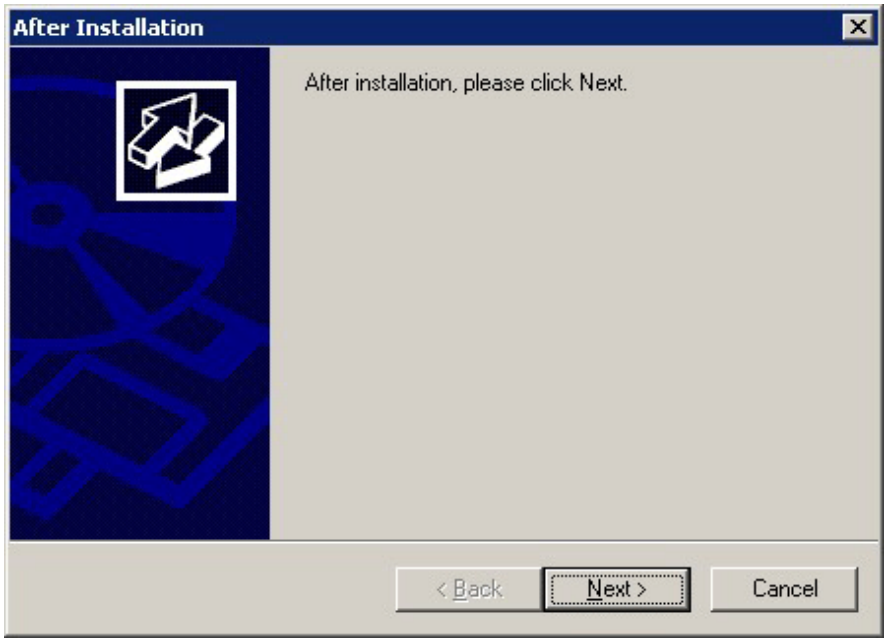
  [Before the installation]

```
CHANGE USER /INSTALL
```

  [After the installation]

```
CHANGE USER /EXECUTE
```

- The following window may be displayed when executing the installation program:



If this window is displayed, perform the following operations:

1. Perform the installation steps until the [InstallShield Wizard Complete] window is displayed.
2. At the window shown above, click [Next].

3. The window shown below is displayed. Click [Finish].

Information

- If a [User Account Control] dialog box is displayed at the start of the installation, click [Yes] to continue processing:

  If [No] is clicked, permission to continue is denied and an [Error] dialog box will be displayed.
  To continue the installation, click [Retry] at the [Error] dialog box. To end the installation, click [Cancel].

- If installation is suspended or processing terminates abnormally, the [Program Compatibility Assistant] dialog box may be displayed. Click [This program installed correctly] and continue operation.

3.2.1 Pre-installation Tasks

This section describes the tasks to be performed before installing the Windows Server Assistant.

Check the disk space

Ensure that there is sufficient disk space to install the Windows Server Assistant.

Refer to "3.1.6 Disk Space Required for Installation" for information on disk space requirements.

hosts file settings

Using an editor, add the IP address and name of the server to be connected to the \System32\drivers\etc\hosts file in the folder where Windows is installed.

Check the installed product and determine the installation method

Using the operation shown below, start Uninstall (middleware), and check the installed products.

In Windows, click [All Programs] or [All apps], then [Fujitsu], and then [Uninstall (middleware)].

If the Windows Server Assistant is already installed, determine the installation method to use:

- Reinstallation
- Multi-version installation

Remove applied updates

If you perform reinstallation as the installation method, remove applied updates using the procedure shown below.

Note

If a product is installed without removing applied updates, the following will occur:

- Performing reinstallation

  If an update with the same update and version number is applied, an error informing you that the update has already been applied is displayed.

1. Display the applied updates

   Execute the following command to display the applied updates:

```
C:\>uam showup
```

2. Remove the updates

   Execute the command below to remove the updates. If an update with the same update number was applied more than once, the updates are removed in order, starting from the highest version number.

```
C:\>uam remove -i update-number
```

3.2.2 Installation in Interactive Mode

The installation procedure is described below.

Point

In interactive mode installation, the default values of the installation information are set. The following setting can be changed for a new installation or a multi-version installation.

- Installation folder

1. Stop the program

If the installation method is the following, the program must be stopped:

- Reinstallation

Before starting the installation, stop the following:

- Mirroring Controller arbitration process

  Execute the mc_arb command in stop mode to stop the Mirroring Controller arbitration process.

  Example

```
> mc_arb stop -M D:\mcarb_dir\arbiter1
```

2. Insert the DVD

Insert the Server Assistant program DVD into the DVD drive.

3. Run the installation

The installation menu is displayed. Click [Installation].

4. Select the product to install

The [Installation product] window is displayed.
Select the product to install, and then click [Next].
If a selected product can only be reinstalled, refer to "6. Check the settings".

Information

If the selected product has been installed, a window for selecting reinstallation or multi-version installation is displayed for each product. Follow the on-screen instructions to select the installation method.

5. Confirm the contents for installation

The [Confirm installation] window will be displayed.

Click [Next] to start the installation.

To modify the settings, select [Modify], and then click [Next]. Follow the on-screen instructions.

6. Check the settings

The [Confirm installation] window will be displayed. Click [Install] to start the installation.

To modify the settings again, click [Back].

7. Completion of installation

The completion window is displayed. Click [Finish].

3.2.3 Installation in Silent Mode

Installation in silent mode can be performed only when the installation method is one of the following:

- New installation
- Multi-version installation

The installation procedure is described below.

See

Refer to the Fujitsu Enterprise Postgres product website for information on installation in silent mode, such as the installation parameters and error messages.

1. Insert the DVD

Insert the Server Assistant program DVD into the DVD drive.

The [Install Menu] window is displayed. Click [Finish].

2. Create an installation parameters CSV file

Create an installation parameters CSV file that uses the following specification format.

```
sectionName, parameterName, value
sectionName, parameterName, value
    :
```

Information

The template for the installation parameters CSV file is "Z:\sample\sample.csv (Z is the drive into which the DVD is inserted.).

3. Start the command prompt

In Windows, right-click [Command Prompt] and then select [Run as administrator].

4. Run the installation

Execute the following command:

```
Z:\>silent.bat c:\temp\inspara.csv
```

Z: The drive into which the DVD is inserted.

c:\temp\inspara.csv: The installation parameter CSV file name.

If the silent installer ends in an error, a message is output to the log file and return values are returned.

3.3 Uninstallation

This section describes how to uninstall the Windows Server Assistant.

Note

- Before uninstalling the product, close the product program.
- Log in using an account that has administrator privileges and then execute the command, or switch to an account that has administrator privileges and then uninstall the product.

Information

- If a [User Account Control] dialog box such as the following is displayed at the start of the uninstallation, click [Yes] to continue processing:

  If [No] is clicked, permission to continue is denied and an [Error] dialog box will be displayed.
  To continue the uninstallation, click [Retry] in the [Error] dialog box. To end the operation, click [Cancel].

- If uninstallation is suspended or processing terminates abnormally, a dialog box of the Program Compatibility Assistant similar to the one shown below may be displayed. Click [This program uninstalled correctly] and continue operation.

3.3.1 Uninstallation in Interactive Mode

The uninstallation procedure is described below.

Information

If an error occurs while the product is being uninstalled, refer to "Uninstall (middleware) Messages" in the Fujitsu Enterprise Postgres product website, and take the required action.

1. Stop the program

Before starting the uninstallation, stop the following:

Mirroring Controller arbitration process

1. Execute the mc_arb command in stop mode to stop the Mirroring Controller arbitration process.

   Example

   ```
   > mc_arb stop -M D:\mcarb_dir\arbiter1
   ```

2. Unregister the Mirroring Controller arbitration process from the Windows service.

   Execute the mc_arb command in unregister mode to unregister the Mirroring Controller arbitration process from the Windows service.

   Example

   ```
   > mc_arb unregister -M D:\mcarb_dir\arbiter1
   ```

3. Delete registrations related to the event log.

   If messages are output to the event log, DLLs are registered in accordance with "Preparatory Tasks for the Output of Error Logs to the Event Log (Windows)" in the Cluster Operation Guide (Database Multiplexing). Delete these registrations so that no unnecessary issues occur.

   - Delete registration for the default event source name

     Example)
     The example below deletes a DLL of a 64-bit product registered under the default event source name. "<*x*>" indicates the product version.

     ```
     > regsvr32 /u "c:\Program Files\Fujitsu\fsepv<x>assistant64\lib\mcarbevent.dll"
     ```

   - Delete registration for other event source names

     DLL registration is performed so that messages output to the event log are output to any event source named by the user.

     Accordingly, it is necessary to delete the DLL registration. Do this for any event source name.

     Example)
     The example below deletes a DLL of a 64-bit product registered under the event source name "Mirroring Controller arbiter1". "<*x*>" indicates the product version.

     ```
     > regsvr32 /u /i:"Mirroring Controller arbiter1" "c:\Program Files\Fujitsu
     \fsepv<x>assistant64\lib\mcarbevent.dll"
     ```

   - If installing multiple versions

     If error logs are set to be output to the event log, use the DLL path name that you took note of previously as explained in "Preparatory Tasks for the Output of Error Logs to the Event Log (Windows)" in the Cluster Operation Guide (Database Multiplexing) to reregister the default event source name.

   Note

   Ensure that you delete the DLLs before the uninstallation. If you perform the uninstallation without doing so, you may not be able to delete the DLLs at a later time.

2. Start Uninstall (middleware)

In Windows, click [All Programs] or [All apps], then [Fujitsu], and then [Uninstall (middleware)].

3. Select the software

Select the product to be uninstalled from [Software Name], and then click [Remove].

4. Start the uninstallation

Click [Uninstall].

5. Finish the uninstallation

The uninstallation completion window will be displayed. Click [Finish].

The installation folder may remain after uninstallation. If it is not required, delete it.

6. Stop Uninstall (middleware)

Click [Close].

3.3.2 Uninstallation in Silent Mode

The uninstallation procedure is described below.

See

Refer to the Fujitsu Enterprise Postgres product website for information on uninstallation in silent mode, such as the error messages.

1. Stop the program

Before starting the uninstallation, stop the following:

Mirroring Controller arbitration process

1. Execute the mc_arb command in stop mode to stop the Mirroring Controller arbitration process.

   Example

   ```
   > mc_arb stop -M D:\mcarb_dir\arbiter1
   ```

2. Unregister the Mirroring Controller arbitration process from the Windows service.

   Execute the mc_arb command in unregister mode to unregister the Mirroring Controller arbitration process from the Windows service.

   Example

   ```
   > mc_arb unregister -M D:\mcarb_dir\arbiter1
   ```

3. Delete registrations related to the event log.

   If messages are output to the event log, DLLs are registered in accordance with "Preparatory Tasks for the Output of Error Logs to the Event Log (Windows)" in the Cluster Operation Guide (Database Multiplexing). Delete these registrations so that no unnecessary issues occur.

   - Delete registration for the default event source name

     Example)
     The example below deletes a DLL of a 64-bit product registered under the default event source name. "<*x*>" indicates the product version.

     ```
     > regsvr32 /u "c:\Program Files\Fujitsu\fsepv<x>assistant64\lib\mcarbevent.dll"
     ```

   - Delete registration for other event source names

     DLL registration is performed so that messages output to the event log are output to any event source named by the user.

     Accordingly, it is necessary to delete the DLL registration. Do this for any event source name.

     Example)
     The example below deletes a DLL of a 64-bit product registered under the event source name "Mirroring Controller arbiter1". "<*x*>" indicates the product version.

```
> regsvr32 /u /i:"Mirroring Controller arbiter1" "c:\Program Files\Fujitsu
\fsepv<x>assistant64\lib\mcarbevent.dll"
```

- If installing multiple versions

  If error logs are set to be output to the event log, use the DLL path name that you took note of previously as explained in "Preparatory Tasks for the Output of Error Logs to the Event Log (Windows)" in the Cluster Operation Guide (Database Multiplexing) to reregister the default event source name.

Note

Ensure that you delete the DLLs before the uninstallation. If you perform the uninstallation without doing so, you may not be able to delete the DLLs at a later time.

2. Start the command prompt

In Windows, right-click [Command Prompt] and then select [Run as administrator].

3. Run the uninstallation

Execute the following command:

The installation folder may remain after uninstallation. If it is not required, delete it.

Example

```
X:> installFolder\suninst.bat
```

X: Drive on which the product is installed

Chapter 4 Setup of the Server Assistant

The Server Assistant is a feature that is installed and used on the arbitration server, so its setup is performed as the arbitration server setup.

See

Refer to "Setting Up Database Multiplexing Mode" in the Cluster Operation Guide (Database Multiplexing) for information on setting up and operating the Mirroring Controller arbitration server.

Appendix A Estimating Memory Requirements

This appendix explains how to estimate the memory.

A.1 Server Assistant Memory Requirements

This section describes the formula for estimating memory requirements for the Server Assistant.

Use the following formula to obtain a rough estimate of memory requirements:

```
Memory usage of the Server Assistant
                                = Peak memory usage of the Mirroring Controller arbitration
processes
                                  + Peak memory usage of the Mirroring Controller commands

Peak memory usage of the Mirroring Controller arbitration processes=100 MB

Peak memory usage of the Mirroring Controller commands=50 MB * Number of commands executed
simultaneously
```

L

Appendix B Procedure when Modifying the JRE Installation

This appendix describes the procedure to follow when modifying the JRE installation.

The JRE, of which the installation destination is specified in the JAVA_HOME environment variable when installing Fujitsu Enterprise Postgres, is used by the Server Assistant.

Therefore, when updating or reinstalling JRE after installing Fujitsu Enterprise Postgres, it is necessary to restart the Mirroring Controller arbitration process, therefore follow the procedure below to modify the JRE installation:

1. Stop the Mirroring Controller arbitration process.

   Refer to the Cluster Operation Guide (Database Multiplexing) for details.

2. Modify the JRE installation.

3. Change the installation environment to be used by Mirroring Controller.

   Set the JAVA_HOME environment variable to the installation destination of JRE 8, and use the mc_update_jre_env command to change the installation environment to be used by the Server Assistant.
   This procedure must be executed by the superuser.

   Example

   /opt/fsepv<*x*>assistant/bin is the installation directory where the Server Assistant is installed.

   ```
   $ su -
   Password:******
   # export JAVA_HOME="Jre8InstallDir"
   # /opt/fsepv<x>assistant/bin/mc_update_jre_env
   ```

4. Start the Mirroring Controller arbitration process.

   Refer to the Cluster Operation Guide (Database Multiplexing) for details.

Index

[C]

Check the disk space....7,16
Check the installed product and determine the installation method 7,16

[D]

Disk Space Required for Installation....6,14

[E]

Estimating Memory Requirements....24
Excluded Software....6,14

[F]

Features that can be Installed....1

[H]

Hardware Environment....6,14
hosts file settings....16

[I]

Installation....14
Installation and Uninstallation of the Linux Server Assistant....3
Installation and Uninstallation of the Windows Server Assistant 13
Installation in Interactive Mode....2,8,17
Installation in Silent Mode....2,10,18
Installation Procedure....1
Installation Types....1

[M]

Multi-Version Installation....1

[N]

New Installation....1

[O]

Operating Environment....3
Operating environment....13

[P]

Pre-installation Tasks....7
Procedure when Modifying the JRE Installation....25

[R]

Reinstallation....1
Related Software....5,13
Remove applied updates....8,17
Required Operating System....3,13
Required Patches....6,14

[S]

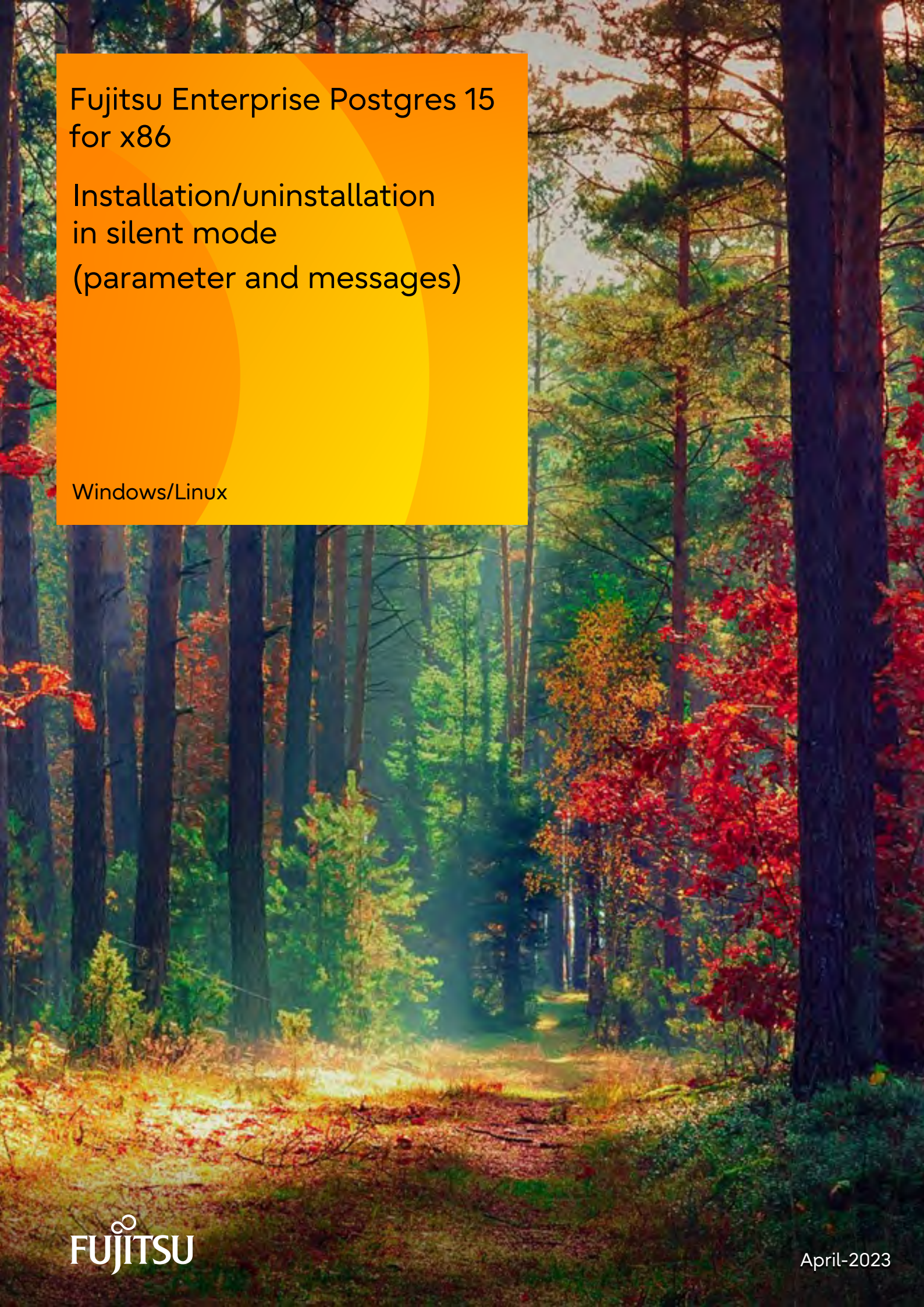
Server Assistant Memory Requirements....24
Setup of the Server Assistant....23
Supported System Environment....6,14

[T]

TCP/IP Protocol....6
TCP/IP protocol....14

[U]

Uninstallation....2,11,19
Uninstallation in Interactive Mode....11,19
Uninstallation in Silent Mode....12,21



Fujitsu Enterprise Postgres 15 for x86

Installation/uninstallation in silent mode (parameter and messages)

Windows/Linux

FUJITSU

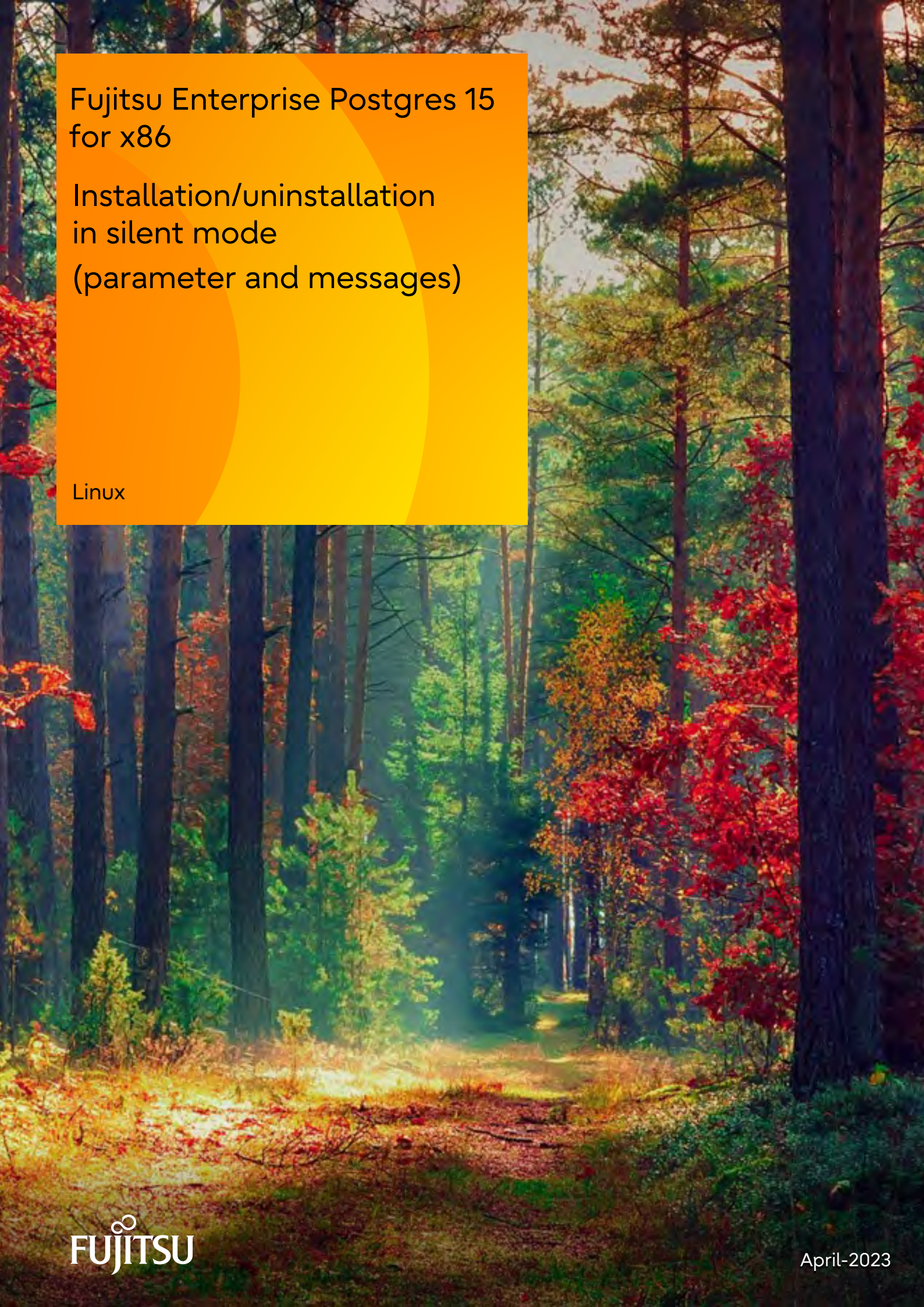
April-2023

Linux >
Windows >

Fujitsu Enterprise Postgres 15 for x86

Installation/uninstallation in silent mode (parameter and messages)

Linux



FUJITSU

April-2023

Preface

Purpose of this document

This document explains the parameters to be specified when installing and uninstalling Fujitsu Enterprise Postgres in silent mode, and explains the installation messages and return values.

Structure of this document

This document is structured as follows:

Chapter 1 Server Feature

Describes how to install and uninstall the Fujitsu Enterprise Postgres server feature in silent mode.

Chapter 2 Client Feature

Describes how to install and uninstall the Fujitsu Enterprise Postgres client feature in silent mode.

Chapter 3 Server Assistant

Describes how to install and uninstall the Fujitsu Enterprise Postgres Server Assistant in silent mode.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

```
Edition 2.0: April 2023
Edition 1.0: January 2022
```

Copyright

Copyright 2018-2023 Fujitsu Limited

Contents

Chapter 1 Server Feature....1
1.1 Installation....1
1.1.1 Specification Format....1
1.1.2 List of Parameters....1
1.1.3 Messages and Return Values....4
1.1.4 CSV File Format....6
1.2 Uninstallation....7
1.2.1 Messages and Return Values....7

Chapter 2 Client Feature....9
2.1 Installation....9
2.1.1 Specification Format....9
2.1.2 List of Parameters....10
2.1.3 Messages and Return Values....12
2.1.4 CSV File Format....14
2.2 Uninstallation....16
2.2.1 Messages and Return Values....16

Chapter 3 Server Assistant....18
3.1 Installation....18
3.1.1 Specification Format....18
3.1.2 List of Parameters....19
3.1.3 Messages and Return Values....20
3.1.4 CSV File Format....22
3.2 Uninstallation....24
3.2.1 Messages and Return Values....24

Chapter 1 Server Feature

This chapter provides specifications for installation and uninstallation of the server feature in silent mode.

1.1 Installation

This chapter provides specifications for installation.

1.1.1 Specification Format

The installation parameters CSV file, which is specified as the argument for the silent installer, has three columns per line in CSV format.

```
sectionName, parameterName, value
sectionName, parameterName, value
    :
```

Enter the following settings in respective columns.

Item	Settings	Optional
sectionName	Specify the section name. There are two types of section names: "installinfo": Set the product information. "parameters": Set the parameter information for this product.	Mandatory
parameterName	Specify the parameter name. Each section has a valid parameter.	Mandatory
value	Specify the value.	Optional

Note

- Blank lines cannot be included.
- Section names and parameter names cannot be omitted.
- Undefined parameters cannot be set in lines where the section name is "installInfo". Also, note that the same parameter cannot be specified multiple times.
- Specify at least one line with the section name "parameters".
- Undefined parameters specified in lines within the "parameters" section will be ignored during execution. Note that when the same parameter is specified multiple times, the settings in the lowest line will be valid.
- The setting values for lines where the section name is "installInfo" may contain alphanumeric characters (at least one), and symbols, excluding double quotation marks (") and commas (,).
- The following characters can be used in the setting for rows with the section name "parameters":

  Alphanumeric characters, hyphens, commas and forward slashes

Information

The template for the installation parameters CSV file is "*mountPoint*/sample/sample.csv".

1.1.2 List of Parameters

This section describes the parameters that can be set for each section.

installInfo section

The parameters that can be set in the installInfo section are shown below.

No.	Type	Parameter		Description
1	Software ID	**Parameter name**	Name	Mandatory. Specify the software identifier. For this software, specify the following value: "Fujitsu Enterprise Postgres"
		Value/Range	String	
		Default value	None	

Information

- In addition to "Name", the following parameters can be used in the "installInfo" section.
  - softwareName
  - OS
  - Version
  - Edition
- Note that specifying a parameter name other than "Name" and the parameters listed above will result in an error.
- The values set in these parameters do not affect silent installation.

Example

```
installInfo,Name,Fujitsu Enterprise Postgres
```

parameters section

The parameters that can be set in the parameters section are shown below.

Note that "<*x*>" indicates the product version.

No.	Type	Parameter		Description
1	Whether to install the server	**Parameter name**	ServerInstallExecute	Optional. Specify whether to run the installation of the server. Y: Install No: Do not install
		Value/Range	Y or N	
		Default value	Y	
2	Installation destination of the server	**Parameter name**	ServerInstallPath	Optional. Specify the installation destination of the software. The root directory (/) cannot be specified.
		Value/Range	Path name	
		Default value	/opt/fsepv<*x*>server64	
3	Whether to install the client (64-bit)	**Parameter name**	Client64InstallExecute	Optional. Specify whether to run the installation of the client (64-bit). Y: Install N: Do not install
		Value/Range	Y or N	
		Default value	Y	

No.	Type	Parameter		Description
4	Installation destination of the client (64-bit)	**Parameter name**	Client64InstallPath	Optional. Specify the installation destination of the client (64-bit). The root directory (/) cannot be specified.
		Value/Range	Path name	
		Default value	/opt/fsepv<*x*>client64	
5	Whether to install WebAdmin	**Parameter name**	WebAdminInstallExecute	Optional. Specify whether to run the installation of WebAdmin in multiserver mode. Y: Install N: Do not install
		Value/Range	Y or N	
		Default value	Y	
6	Installation destination of WebAdmin	**Parameter name**	WebAdminInstallPath	Optional. Specify the installation destination of WebAdmin. The root directory (/) cannot be specified.
		Value/Range	Path name	
		Default value	/opt/fsepv<*x*>webadmin	
7	Whether to set up WebAdmin	**Parameter name**	WebSetupExecute	Optional. Specify whether to set up WebAdmin. Y: Setup N: Do not set up This will be ignored if the WebAdmin is not installed.
		Value/Range	Y or N	
		Default value	Y	
8	Web server port number	**Parameter name**	WebPortNumber1	Optional. Specify the port number of the Web server. This will be ignored if the WebAdmin is not installed.
		Value/Range	1024 to 32767	
		Default value	27515	
9	WebAdmin internal port number	**Parameter name**	WebPortNumber2	Optional. Specify the WebAdmin internal port number. This will be ignored if the WebAdmin is not installed.
		Value/Range	1024 to 32767	
		Default value	27516	
10	Whether to install Pgpool-II	**Parameter name**	Pgpool2InstallExecute	Specify whether to run the installation of the Pgpool-II. Y: Install N: Do not install
		Value/Range	Y or N	
		Default value	N	
11	Installation destination of Pgpool-II	**Parameter name**	Pgpool2InstallPath	Optional. Specify the installation destination of Pgpool-II. The root directory (/) cannot be specified.
		Value/Range	Path name	
		Default value	/opt/fsepv<*x*>pgpool-II	

Example

```
parameters,ServerInstallExecute,Y
parameters,ServerInstallPath,
parameters,Client64InstallExecute,Y
parameters,Client64InstallPath,
parameters,WebAdminInstallExecute,Y
```

```
parameters,WebAdminInstallPath,
parameters,WebSetupExecute,Y
parameters,WebPortNumber1,27515
parameters,WebPortNumber2,27516
parameters,Pgpool2InstallExecute,Y
parameters,Pgpool2InstallPath,
```

1.1.3 Messages and Return Values

Messages are output when errors are detected during parametric analysis.

If an error is detected during installation of the product, a message is output to the log:

Log file

In "*xyz*" below, "*x*" is the product version, "*z*" is the product level, and for this release, "*y*" is 0.

```
var/log/fsep_SERVER64_media_xyz_install.log
```

Messages and return values

CSV file errors

The following messages are output if errors are detected while parsing CSV files.

Return value	Message	Explanation and actions
4	CSV file error:code = 1, Invalid CSV error.	There is an error in the specification format of the CSV file.
4	CSV file error:code = 2, installInfo/Name is required.	Either installInfo or the Name parameter has not been specified.
4	CSV file error:code = 3, Invalid installInfo key.	There is an error in the installInfo specification. Or the section name is invalid.
4	CSV file error:code = 4, Duplicated installInfo key.	The same parameter has been defined more than once in installInfo.
4	CSV file error:code = 5, Invalid character length.	No setting value is specified, or the specified string is too long.
4	CSV file error:code = 6, Invalid character format or encoding.	An invalid character has been specified in the installation parameters CSV file.
4	CSV file error:code = 8, Parameter is required.	There is no line in the "parameters" section.
20	The input file does not exist.	The input file does not exist.
21	The value of @1@ is incorrect, the value is @2@.	The value is incorrect. Specify the correct value. The parameter name is displayed in @1@. The specified parameter is displayed in @2@.
22	The value same at @1@ and @2@ is specified.	The same value is specified in different parameters. Specify different values. The parameter name is displayed in @1@ and @2@.
23	The @2@ of @1@ already exists.	The path already exists. Specify a different path. The parameter name is displayed in @1@. The specified value is displayed in @2@.

Return value	Message	Explanation and actions
26	Port number @1@ is already used in the service file(/etc/services).	The port number is already being used. Specify an unused port number. The port number is displayed in @1@.
28	There is no product for installation.	No product was specified for installation. Specify "Y" for at least one of the following parameters: - ServerInstallExecute - Client64InstallExecute - WebAdminInstallExecute - Pgpool2InstallExecute
29	USAGE : silent.sh inputfile	The argument specified in the command is incorrect. Specify the correct argument.

Product installer errors

The following return values are returned when errors occur while the product installer is running.

Return value	Explanation
0	Completed successfully.
3	Failed to setup WebAdmin.
13	Processing was interrupted.
14	The resources required for installation do not exist on the media.
15	The resources required for installation do not exist.
16	Unable to change the permission because the directory does not exist.
17	Failed to install because a required package is missing.
19	A process is running.
30	Failed to copy the installation material.
32	The same version of the same product is already installed.
60	An error occurred in Uninstall (middleware).
61	Failed to install Uninstall (middleware).
62	Uninstall (middleware) was started.
63	The uninstallation cannot be performed either because Uninstall (middleware) does not exist or because there are no execution privileges.
83	Failed to install FJQSS or register information.
85	Different editions of the same generation are installed.
86	A newer version of the product is already installed.
88	Exclusive product is already installed.
99	A system error occurred.
106	The command was not executed with administrator privileges.
110	License registration failed.
129	Unsupported operating system.

Return value	Explanation
200	Failed to install of package.
201	A conflicting package is installed.

1.1.4 CSV File Format

The format of CSV files is based on RFC4180, with the following specifications.

Records

- Separate each record with a "CRLF" newline (operation is not guaranteed with only a "CR" or "LF" newline).
- Specify a newline at the end of a file.
- Separate each field within a record with a halfwidth comma ",".

Format	Record		
aaa,bbb,ccc	aaa	bbb	ccc

- If several commas are entered in succession, or if a comma precedes a newline, the data following the comma is regarded as empty.

Format	Record		
aaa,,ccc	aaa		ccc
aaa,bbb,	aaa	bbb	

- Headers cannot be specified.

Format	Record		
	field1	field2	field 3
aaa,bbb,ccc	aaa	bbb	ccc

Double quotation marks

- Enclose fields that contain newlines, double quotation marks, or commas in double quotation marks.

Format	Record		
"aaa","bb b","ccc"	aaa	bb b	ccc
"aaa","bb,b","ccc"	aaa	bb,b	ccc

- If a double quotation mark is used in a field enclosed with double quotation marks, the first double quotation mark is regarded as an escape character.

Format	Record		
"aaa","bb""b","ccc"	aaa	bb"b	ccc

Note

An error occurs in the following cases:

- Each space, tab, or whitespace character is recognized as one character, and will cause a parameter error.
- A space is entered before or after a field enclosed with double quotation marks.

```
"zzz ", "yyy ", " xxx "
```

- The number of fields differs between records.

```
aaa,bbb
aaa,bbb,ccc
aaa,bbb,ccc,ddd
```

- Fields enclosed with double quotation marks and fields not enclosed with double quotation marks are both used.

```
aaa,"bbb",ccc
"xxx",yyy,"zzz"
```

- The double quotation escape characters are not positioned correctly.

```
"aaa","bb""""b","ccc"
```

- Double quotation marks enclose the entire field.

```
"aaa,bbb,ccc"
```

1.2 Uninstallation

This chapter provides specifications for uninstallation.

1.2.1 Messages and Return Values

The uninstaller result is output to the log file.

Log file

In "*xyz*" below, "*x*" is the product version, "*z*" is the product level, and for this release, "*y*" is 0.

Server product

```
/var/log/fsep_SERVER64_xyz_uninstall.log
```

WebAdmin

```
/var/log/fsep_WEBADMIN_xyz_uninstall.log
```

Pgpool-II

```
/var/log/fsep_PGPOOL2_xyz_uninstall.log
```

Return values

The following return values are output:

Return values	Meaning
0	Uninstallation was successful.
3	WebAdmin unsetup failed.
13	Processing was interrupted.
15	The resources required for uninstallation do not exist.
19	A process is running.
60	An error occurred in Uninstall (middleware).
62	Uninstall (middleware) was started.

Return values	Meaning
63	Uninstall (middleware) has not been installed, or there are no execution privileges.
83	Failed to delete the information from FJQSS.
99	A system error occurred.
100	The command argument is invalid.
106	The command was not executed with administrator privileges.
200	Failed to uninstall of package.

Chapter 2 Client Feature

This chapter provides specifications for installation and installation of the client feature in silent mode.

Note that "<*x*>" in paths indicates the product version.

2.1 Installation

This chapter provides specifications for installation.

2.1.1 Specification Format

The installation parameters CSV file, which is specified as the argument for the silent installer, has three columns per line in CSV format.

```
sectionName, parameterName, value
sectionName, parameterName, value
    :
```

Enter the following settings in respective columns.

Item	Settings	Optional
sectionName	Specify the section name. There are two types of section names: "installInfo": Set the product information. "parameters": Set the parameter information for this product.	Mandatory
parameterName	Specify the parameter name. Each section has a valid parameter.	Mandatory
value	Specify the value.	Optional

Note

- Blank lines cannot be included.
- Section names and parameter names cannot be omitted.
- Undefined parameters cannot be set in lines where the section name is "installInfo". Also, note that the same parameter cannot be specified multiple times.
- Specify at least one line with the section name "parameters".
- Undefined parameters specified in lines within the "parameters" section will be ignored during execution. Note that when the same parameter is specified multiple times, the settings in the lowest line will be valid.
- The setting values for lines where the section name is "installInfo" may contain alphanumeric characters (at least one), and symbols, excluding double quotation marks (") and commas (,).
- Do not use the following halfwidth characters in the setting values for lines with the section name "parameters".

W Windows: # $ & ' = ^ ~ | @ ` [ ] ; < > / tab

L Linux: " # $ & = ' ( ) ^ ~ \ | @ ` [ ] { } ; : < > tab

Information

The template for the installation parameters CSV file is as follows:

- W Windows client: *Z*:\sample\sample_windows.csv (*Z* is the drive into which the DVD is inserted.)

- Linux client: *mountPoint*/sample/sample_linux.csv

2.1.2 List of Parameters

This section describes the parameters that can be set for each section.

installInfo section

The parameters that can be set in the installInfo section are shown below.

No.	Type	Parameter		Description
1	Software ID	**Parameter name**	Name	Mandatory. Specify the software identifier. For this software, specify the following value: "Fujitsu Enterprise Postgres"
		Value/Range	String	
		Default value	None	

Information

- In addition to "Name", the following parameters can be used in the "installInfo" section.
  - softwareName
  - OS
  - Version
  - Edition
- Note that specifying a parameter name other than "Name" and the parameters listed above will result in an error.
- The values set in these parameters do not affect silent installation.

Example

```
installInfo,Name,Fujitsu Enterprise Postgres
```

parameters section

The parameters that can be set in the parameters section are shown below.

No.	Type	Parameter		Description
1	Whether to install the client (32-bit)	**Parameter name**	Client32InstallExecute	Optional. Specify whether to run the installation of the client (32-bit). Y: Install N: Do not install This parameter is ignored on Linux.
		Value/Range	Y or N	
		Default value	Y	
2	Installation destination of the client (32-bit)	**Parameter name**	Client32InstallPath	Optional. Specify the installation destination of the client (32-bit).
		Value/Range	Path name	

No.	Type	Parameter		Description
		Default value	W Windows(32-bit) %ProgramFiles%\Fujitsu\fsepv<*x*>client32 Windows(64-bit) %ProgramFiles(x86)%\Fujitsu\fsepv<*x*>client32 L Linux None.	W A root folder (such as C:\) cannot be specified. L This parameter is ignored on Linux.
3	Whether to install the client (64-bit)	**Parameter name**	Client64InstallExecute	Optional. Specify whether to run the installation of the client (64-bit). Y: Install N: Do not install W This parameter is ignored on 32-bit operating systems.
		Value/Range	Y or N	
		Default value	Y	
4	Installation destination of the client (64-bit)	**Parameter name**	Client64InstallPath	Optional. Specify the installation destination of the client (64-bit). W This parameter is ignored on 32-bit operating systems. A root folder (such as C:\) cannot be specified. L A root directory cannot be specified.
		Value/Range	Path name	
		Default value	W Windows(64-bit) %ProgramFiles%\Fujitsu\fsepv<*x*>client64 L Linux /opt/fsepv<*x*>client64	
5	Whether to install Pgpool-II	**Parameter name**	Pgpool2InstallExecute	Specify whether to run the installation of the Pgpool-II. Y: Install N: Do not install W This parameter is ignored on Windows.
		Value/Range	Y or N	
		Default value	N	
6	Installation destination of Pgpool-II	**Parameter name**	Pgpool2InstallPath	Optional. Specify the installation destination of Pgpool-II. W This parameter is ignored on Windows. L A root directory cannot be specified.
		Value/Range	Path name	
		Default value	/opt/fsepv<*x*>pgpool-II	

Example

```
parameters,Client64InstallExecute,Y
parameters,Client64InstallPath,
parameters,Pgpool2InstallExecute,Y
parameters,Pgpool2InstallPath,
```

2.1.3 Messages and Return Values

Messages are output when errors are detected during parametric analysis.

If an error occurs during installation of the product, a message is output to the log.

Log file

In "*xyz*" below, "*x*" is the product version, "*z*" is the product level, and for this release, "*y*" is 0.

Windows

```
%TEMP%\fsep_CLIENT_media_xyz.log
```

Linux

```
/var/log/fsep_CLIENT_media_xyz_install.log
```

Messages and return values

CSV file errors

The following messages are output if errors are detected while parsing CSV files.

Return value	Message	Explanation and actions
4	CSV file error:code = 1, Invalid CSV error.	There is an error in the specification format of the CSV file.
4	CSV file error:code = 2, installInfo/Name is required.	There is an error in the specification format of the CSV file.
4	CSV file error:code = 3, Invalid installInfo key.	There is an error in the installInfo specification. Or the section name is invalid.
4	CSV file error:code = 4, Duplicated installInfo key.	The same parameter has been defined more than once in installInfo.
4	CSV file error:code = 5, Invalid character length.	No setting value has been specified, or the specified string is too long.
4	CSV file error:code = 6, Invalid character format or encoding.	An invalid character has been specified in the installation parameters CSV file.
4	CSV file error:code = 8, Parameter is required.	The "parameters" section has no lines.
20	The input file does not exist.	The input file does not exist.
21	The value of @1@ is incorrect,the value is @2@.	The value is incorrect. Specify the correct value. The parameter name is displayed in @1@. The specified value is displayed in @2@.
22	The value same at @1@ and @2@ is specified.	The same value is specified in different parameters. Specify different values. The parameter name is displayed in @1@ and @2@.

Return value	Message	Explanation and actions
23	The @2@ of @1@ already exists.	The path already exists. Specify a different path. The parameter name is displayed in @1@. The specified value is displayed in @2@.
24	Drive @2@ of @1@ does not exist.	The drive does not exist. Specify an existing drive. The parameter name is displayed in @1@. The specified value is displayed in @2@.
28	There is no product for installation.	There is no product to be installed. Specify "Y" for at least one of the following parameters: Windows - Client32InstallExecute - Client64InstallExecute Linux - Client64InstallExecute - Pgpool2InstallExecute
29	Windows USAGE : silent.bat inputfile Linux USAGE : silent.sh inputfile	The argument specified in the command is incorrect. Specify the correct argument.

Product installer errors

The following return values are returned when errors occur while the product installer is running.

Windows

Return value	Explanation
0	Completed successfully.
14	The resources required for installation do not exist on the media.
32	The same version of the same product is already installed.
60	An error occurred in Uninstall (middleware).
61	Failed to install Uninstall (middleware).
62	Uninstall (middleware) was started.
74	Failed to write to the registry.
79	Failed to load the registry.
85	Different editions of the same generation are installed.
86	A newer version of the product is already installed.
88	Exclusive product is already installed.
129	The operating system is not supported.
142	Insufficient disk space at the installation destination.
144	The command was not executed with administrator privileges.
200	Failed to install of package.

Return value	Explanation
201	A conflicting package is installed.
220	Failed to create a folder in the installation folder.
221	Failed to create files in the installation folder.
251	An error occurred when running a system function.
255	An internal inconsistency occurred.

L Linux

Return value	Explanation
0	Completed successfully.
13	Processing was interrupted.
14	The resources required for installation do not exist on the media.
15	The resources required for installation do not exist.
16	Unable to change the permission because the directory does not exist.
17	Failed to install because a required package is missing.
19	A process is running.
30	Failed to copy the installation resources.
32	The same version of the same product is already installed.
60	An error occurred in Uninstall (middleware).
61	Failed to install Uninstall (middleware).
62	Uninstall (middleware) was started.
63	The uninstallation cannot be performed either because Uninstall (middleware) does not exist or because there are no execution privileges.
85	Different editions of the same generation are installed.
86	A newer version of the product is already installed.
88	Exclusive product is already installed.
99	A system error occurred.
106	The command was not executed with administrator privileges.
129	Unsupported operating system.
200	Failed to install of package.
201	A conflicting package is installed.

2.1.4 CSV File Format

The format of CSV files is based on RFC4180, with the following specifications.

Records

- Separate each record with a "CRLF" newline (operation is not guaranteed with only a "CR" or "LF" newline).
- Specify a newline at the end of a file.
- Separate each field within a record with a halfwidth comma ",".

Format	Record		
aaa,bbb,ccc	aaa	bbb	ccc

- If several commas are entered in succession, or if a comma precedes a newline, the data following the comma is regarded as empty.

Format	Record		
aaa,,ccc	aaa		ccc
aaa,bbb,	aaa	bbb	

- You cannot specify a header.

Format	Record		
	field1	field2	field 3
aaa,bbb,ccc	aaa	bbb	ccc

Double quotation marks

- Enclose fields that contain newlines, double quotation marks, or commas in double quotation marks.

Format	Record		
"aaa","bb b","ccc"	aaa	bb b	ccc
"aaa","bb,b","ccc"	aaa	bb,b	ccc

- If a double quotation mark is used in a field enclosed with double quotation marks, the first double quotation mark is regarded as an escape character.

Format	Record		
"aaa","bb""b","ccc"	aaa	bb"b	ccc

An error occurs in the following cases:

- Each space, tab, or whitespace character is recognized as one character, and will cause a parameter error.
- A space is entered before or after a field enclosed with double quotation marks.

```
"zzz ", "yyy ", " xxx "
```

- The number of fields differs between records.

```
aaa,bbb
aaa,bbb,ccc
aaa,bbb,ccc,ddd
```

- Fields enclosed with double quotation marks and fields not enclosed with double quotation marks are both used.

```
aaa,"bbb",ccc
"xxx",yyy,"zzz"
```

- The double quotation escape characters are not positioned correctly.

```
"aaa","bb"""b","ccc"
```

- Double quotation marks enclose the entire field.

```
"aaa,bbb,ccc"
```

2.2 Uninstallation

This chapter provides specifications for uninstallation.

2.2.1 Messages and Return Values

The uninstaller result is output to the log file.

Log file

In "*xyz*" below, "*x*" is the product version, "*z*" is the product level, and for this release, "*y*" is 0.

Windows

64-bit product

```
%TEMP%\fsep_CLIENT64_xyz_uninstall.log
```

32-bit product

```
%TEMP%\fsep_CLIENT32_xyz_uninstall.log
```

Linux

64-bit product

```
/var/log/fsep_CLIENT64_xyz_uninstall.log
```

Pgpool-II

```
/var/log/fsep_PGPOOL2_xyz_uninstall.log
```

Return values

The following return values are output:

Windows

Return values	Meaning
0	Uninstallation was successful.
64	Failed to install because "Uninstall (middleware)" is running.
74	Failed to write product information (updating registry).
79	Failed to read product information (viewing registry).
144	The command was not executed with administrator privileges.
200	Failed to uninstall of package.
251	An error occurred when running a system function.
255	An internal inconsistency occurred.

Linux

Return values	Meaning
0	Uninstallation was successful.

Return values	Meaning
13	Processing was interrupted.
15	The resources required for uninstallation do not exist.
19	A process is running.
60	An error occurred in Uninstall (middleware).
62	Uninstall (middleware) was started.
63	Uninstall (middleware) has not been installed, or there are no execution privileges.
99	A system error occurred.
100	The command argument is invalid.
106	The command was not executed with administrator privileges.
200	Failed to uninstall of package.

Chapter 3 Server Assistant

This chapter provides specifications for installation and uninstallation of the Server Assistant in silent mode.

Note that "<*x*>" in paths indicates the product version.

3.1 Installation

This chapter provides specifications for installation.

3.1.1 Specification Format

The installation parameters CSV file, which is specified as the argument for the silent installer, has three columns per line in CSV format.

```
sectionName, parameterName, value
sectionName, parameterName, value
    :
```

Enter the following settings in respective columns.

Item	Settings	Optional
sectionName	Specify the section name. There are two types of section names: "installInfo": Set the product information. "parameters": Set the parameter information for this product.	Mandatory
parameterName	Specify the parameter name. Each section has a valid parameter.	Mandatory
value	Specify the value.	Optional

Note

- Blank lines cannot be included.
- Section names and parameter names cannot be omitted.
- Undefined parameters cannot be set in lines where the section name is "installInfo". Also, note that the same parameter cannot be specified multiple times.
- Specify at least one line with the section name "parameters".
- Undefined parameters specified in lines within the "parameters" section will be ignored during execution. Note that when the same parameter is specified multiple times, the settings in the lowest line will be valid.
- The setting values for lines where the section name is "installInfo" may contain alphanumeric characters (at least one), and symbols, excluding double quotation marks (") and commas (,).
- Do not use the following halfwidth characters in the setting values for lines with the section name "parameters".

  Windows: # $ & ' = ^ ~ | @ ` [ ] ; < > / tab

  Linux: " # $ & = ' ( ) ^ ~ \ | @ ` [ ] { } ; : < > tab

Information

The template for the installation parameters CSV file is as follows:

- Windows Server Assistant: *Z*:\sample\sample.csv (*Z* is the drive into which the DVD is inserted.)

- Linux Server Assistant: *mountPoint*/sample/sample.csv

3.1.2 List of Parameters

This section describes the parameters that can be set for each section.

installInfo section

The parameters that can be set in the installInfo section are shown below.

No.	Type	Parameter		Description
1	Software ID	**Parameter name**	Name	Mandatory. Specify the software identifier. For this software, specify the following value: "Fujitsu Enterprise Postgres"
		Value/Range	String	
		Default value	None	

Information

- In addition to "Name", the following parameters can be used in the "installInfo" section.
  - softwareName
  - OS
  - Version
  - Edition
- Note that specifying a parameter name other than "Name" and the parameters listed above will result in an error.
- The values set in these parameters do not affect silent installation.

Example

```
installInfo,Name,Fujitsu Enterprise Postgres
```

parameters section

The parameters that can be set in the parameters section are shown below.

No.	Type	Parameter		Description
1	Whether to install the server assistant	**Parameter name**	ArbiterInstallExecute	Optional. Specify whether to run the installation of the Server Assistant. Y: Install N: Do not install
		Value/Range	Y or N	
		Default value	Y	
2	Installation destination of the server assistant	**Parameter name**	ArbiterInstallPath	Optional. Specify the installation destination of the Server Assistant. W A root folder (such as C:\) cannot be specified. L A root directory cannot be specified.
		Value/Range	Path name	
		Default value	W Windows %ProgramFiles%\Fujitsu\fsepv<*x*>assistant64	

No.	Type	Parameter		Description
			L Linux /opt/fsepv<*x*>assistant	

Example

```
parameters,ArbiterInstallExecute,Y
parameters,ArbiterInstallPath,
```

3.1.3 Messages and Return Values

Messages are output when errors are detected during parametric analysis.

If an error occurs during installation of the product, a message is output to the log.

Log file

In "*xyz*" below, "*x*" is the product version, "*z*" is the product level, and for this release, "*y*" is 0.

W Windows

```
%TEMP%\fsep_SERVER_ASSISTANT_media_xyz.log
```

L Linux

```
/var/log/fsep_SERVER_ASSISTANT_media_xyz_install.log
```

Messages and return values

CSV file errors

The following messages are output if errors are detected while parsing CSV files.

Return value	Message	Explanation and actions
4	CSV file error:code = 1, Invalid CSV error.	There is an error in the specification format of the CSV file.
4	CSV file error:code = 2, installInfo/Name is required.	There is an error in the specification format of the CSV file.
4	CSV file error:code = 3, Invalid installInfo key.	There is an error in the installInfo specification. Or the section name is invalid.
4	CSV file error:code = 4, Duplicated installInfo key.	The same parameter has been defined more than once in installInfo.
4	CSV file error:code = 5, Invalid character length.	No setting value has been specified, or the specified string is too long.
4	CSV file error:code = 6, Invalid character format or encoding.	An invalid character has been specified in the installation parameters CSV file.
4	CSV file error:code = 8, Parameter is required.	The "parameters" section has no lines.
20	The input file does not exist.	The input file does not exist.
21	The value of @1@ is incorrect,the value is @2@.	The value is incorrect. Specify the correct value. The parameter name is displayed in @1@. The specified value is displayed in @2@.
22	The value same at @1@ and @2@ is specified.	The same value is specified in different parameters. Specify different values.

Return value	Message	Explanation and actions
		The parameter name is displayed in @1@ and @2@.
23	The @2@ of @1@ already exists.	The path already exists. Specify a different path. The parameter name is displayed in @1@. The specified value is displayed in @2@.
24	Drive @2@ of @1@ does not exist.	The drive does not exist. Specify an existing drive. The parameter name is displayed in @1@. The specified value is displayed in @2@.
28	There is no product for installation.	There is no product to be installed. Specify "Y" for the following parameters: - ArbiterInstallExecute
29	Windows USAGE : silent.bat inputfile Linux USAGE : silent.sh inputfile	The argument specified in the command is incorrect. Specify the correct argument.

Product installer errors

The following return values are returned when errors occur while the product installer is running.

Windows

Return value	Explanation
0	Completed successfully.
14	The resources required for installation do not exist on the media.
32	The same version of the same product is already installed.
60	An error occurred in Uninstall (middleware).
61	Failed to install Uninstall (middleware).
62	Uninstall (middleware) was started.
74	Failed to write to the registry.
79	Failed to load the registry.
83	Failed to install FJQSS or register information.
86	A newer version of the product is already installed.
88	Exclusive product is already installed.
129	The operating system is not supported.
142	Insufficient disk space at the installation destination.
144	The command was not executed with administrator privileges.
200	Failed to install of package.
201	A conflicting package is installed.
220	Failed to create a folder in the installation folder.
221	Failed to create files in the installation folder.
251	An error occurred when running a system function.

Return value	Explanation
255	An internal inconsistency occurred.

Linux

Return value	Explanation
0	Completed successfully.
13	Processing was interrupted.
14	The resources required for installation do not exist on the media.
15	The resources required for installation do not exist.
16	Unable to change the permission because the directory does not exist.
17	Failed to install because a required package is missing.
19	A process is running.
30	Failed to copy the installation resources.
32	The same version of the same product is already installed.
60	An error occurred in Uninstall (middleware).
61	Failed to install Uninstall (middleware).
62	Uninstall (middleware) was started.
63	The uninstallation cannot be performed either because Uninstall (middleware) does not exist or because there are no execution privileges.
83	Failed to install FJQSS or register information.
86	A newer version of the product is already installed.
88	Exclusive product is already installed.
99	A system error occurred.
106	The command was not executed with administrator privileges.
129	Unsupported operating system.
200	Failed to install of package.
201	A conflicting package is installed.

3.1.4 CSV File Format

The format of CSV files is based on RFC4180, with the following specifications.

Records

- Separate each record with a "CRLF" newline (operation is not guaranteed with only a "CR" or "LF" newline).
- Specify a newline at the end of a file.
- Separate each field within a record with a halfwidth comma ",".

Format	Record		
aaa,bbb,ccc	aaa	bbb	ccc

- If several commas are entered in succession, or if a comma precedes a newline, the data following the comma is regarded as empty.

Format	Record		
aaa,,ccc	aaa		ccc
aaa,bbb,	aaa	bbb	

- You cannot specify a header.

Format	Record		
	field1	field2	field 3
aaa,bbb,ccc	aaa	bbb	ccc

Double quotation marks

- Enclose fields that contain newlines, double quotation marks, or commas in double quotation marks.

Format	Record		
"aaa","bb b","ccc"	aaa	bb b	ccc
"aaa","bb,b","ccc"	aaa	bb,b	ccc

- If a double quotation mark is used in a field enclosed with double quotation marks, the first double quotation mark is regarded as an escape character.

Format	Record		
"aaa","bb""b","ccc"	aaa	bb"b	ccc

Note

An error occurs in the following cases:

- Each space, tab, or whitespace character is recognized as one character, and will cause a parameter error.
- A space is entered before or after a field enclosed with double quotation marks.

```
"zzz ", "yyy ", " xxx "
```

- The number of fields differs between records.

```
aaa,bbb
aaa,bbb,ccc
aaa,bbb,ccc,ddd
```

- Fields enclosed with double quotation marks and fields not enclosed with double quotation marks are both used.

```
aaa,"bbb",ccc
"xxx",yyy,"zzz"
```

- The double quotation escape characters are not positioned correctly.

```
"aaa","bb"""b","ccc"
```

- Double quotation marks enclose the entire field.

```
"aaa,bbb,ccc"
```

3.2 Uninstallation

This chapter provides specifications for uninstallation.

3.2.1 Messages and Return Values

The uninstaller result is output to the log file.

Log file

In "*xyz*" below, "*x*" is the product version, "*z*" is the product level, and for this release, "*y*" is 0.

Windows

```
%TEMP%\fsep_SERVER_ASSISTANT_xyz_uninstall.log
```

Linux

```
/var/log/fsep_SERVER_ASSISTANT_xyz_uninstall.log
```

Return values

The following return values are output:

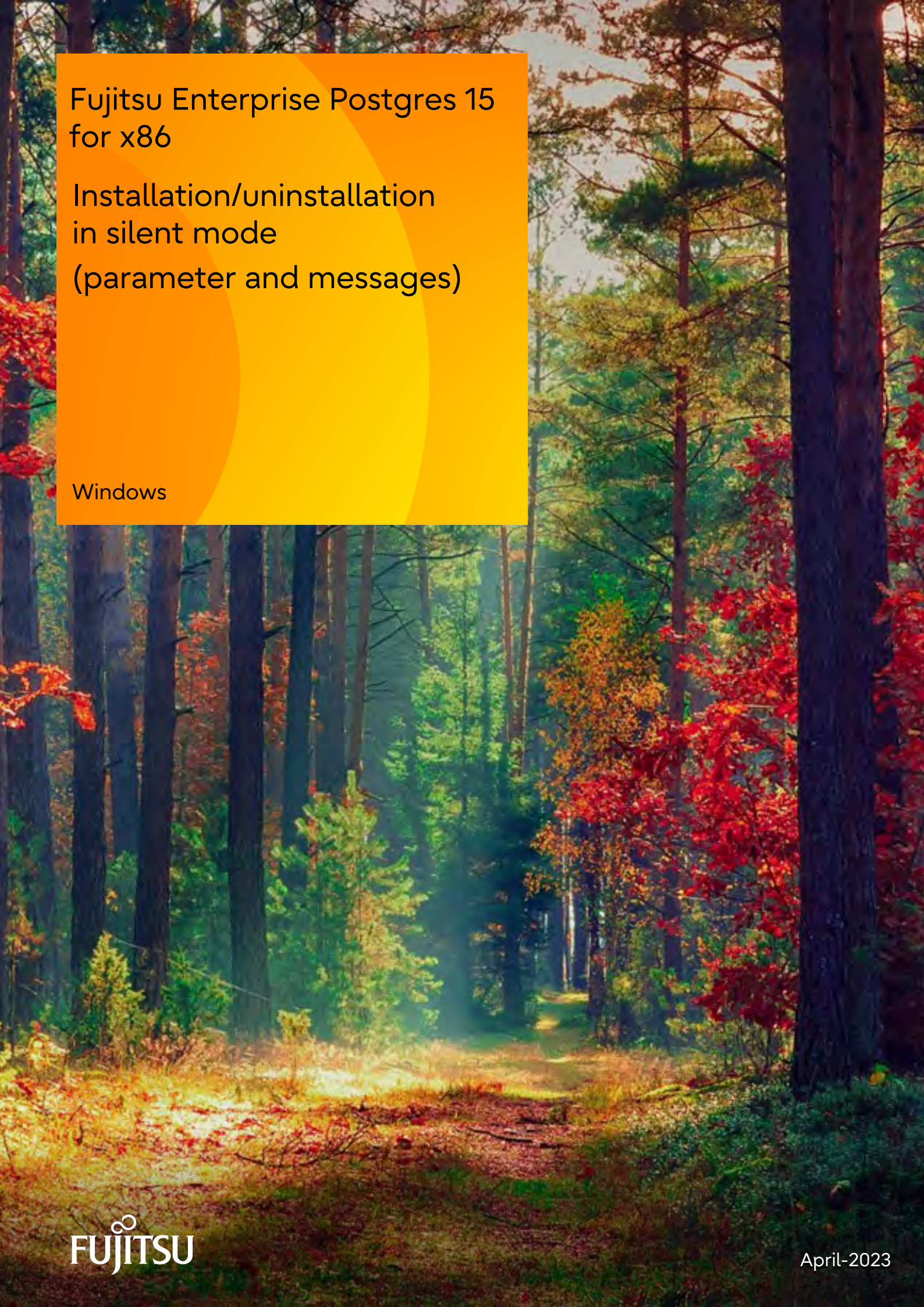
Windows

Return values	Meaning
0	Uninstallation was successful.
64	Failed to install because "Uninstall (middleware)" is running.
74	Failed to write product information (updating registry).
79	Failed to read product information (viewing registry).
83	Failed to delete the information from FJQSS.
144	The command was not executed with administrator privileges.
200	Failed to uninstall of package.
251	An error occurred when running a system function.
255	An internal inconsistency occurred.

Linux

Return values	Meaning
0	Uninstallation was successful.
13	Processing was interrupted.
15	The resources required for uninstallation do not exist.
19	A process is running.
60	An error occurred in Uninstall (middleware).
62	Uninstall (middleware) was started.
63	Uninstall (middleware) has not been installed, or there are no execution privileges.
83	Failed to delete the information from FJQSS.
99	A system error occurred.
100	The command argument is invalid.

Return values	Meaning
106	The command was not executed with administrator privileges.
200	Failed to uninstall of package.



Fujitsu Enterprise Postgres 15 for x86

Installation/uninstallation in silent mode (parameter and messages)

Windows

FUJITSU

April-2023

Preface

Purpose of this document

This document explains the parameters to be specified when installing and uninstalling Fujitsu Enterprise Postgres in silent mode, and explains the installation messages and return values.

Structure of this document

This document is structured as follows:

Chapter 1 Server Feature

Describes how to install and uninstall the Fujitsu Enterprise Postgres server feature in silent mode.

Chapter 2 Client Feature

Describes how to install and uninstall the Fujitsu Enterprise Postgres client feature in silent mode.

Chapter 3 Server Assistant

Describes how to install and uninstall the Fujitsu Enterprise Postgres Server Assistant in silent mode.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

```
Edition 2.0: April 2023
Edition 1.0: February 2022
```

Copyright

Copyright 2018-2023 Fujitsu Limited

Contents

Chapter 1 Server Feature 1
1.1 Installation 1
1.1.1 Specification Format 1
1.1.2 List of Parameters 2
1.1.3 Messages and Return Values 4
1.1.4 CSV File Format 6
1.2 Uninstallation 7
1.2.1 Messages and Return Values 7

Chapter 2 Client Feature 9
2.1 Installation 9
2.1.1 Specification Format 9
2.1.2 List of Parameters 10
2.1.3 Messages and Return Values 12
2.1.4 CSV File Format 14
2.2 Uninstallation 16
2.2.1 Messages and Return Values 16

Chapter 3 Server Assistant 18
3.1 Installation 18
3.1.1 Specification Format 18
3.1.2 List of Parameters 19
3.1.3 Messages and Return Values 20
3.1.4 CSV File Format 22
3.2 Uninstallation 24
3.2.1 Messages and Return Values 24

Chapter 1 Server Feature

This chapter provides specifications for installation and uninstallation of the server feature in silent mode.

1.1 Installation

This chapter provides specifications for installation.

1.1.1 Specification Format

The installation parameters CSV file, which is specified as the argument for the silent installer, has three columns per line in CSV format.

```
sectionName, parameterName, value
sectionName, parameterName, value
    :
```

Enter the following settings in respective columns.

Item	Settings	Optional
sectionName	Specify the section name. There are two types of section names: "installinfo": Set the product information. "parameters": Set the parameter information for this product.	Mandatory
parameterName	Specify the parameter name. Each section has a valid parameter.	Mandatory
value	Specify the value.	Optional

Note

- Blank lines cannot be included.
- Section names and parameter names cannot be omitted.
- Undefined parameters cannot be set in lines where the section name is "installInfo". Also, note that the same parameter cannot be specified multiple times.
- Specify at least one line with the section name "parameters".
- Undefined parameters specified in lines within the "parameters" section will be ignored during execution. Note that when the same parameter is specified multiple times, the settings in the lowest line will be valid.
- The setting values for lines where the section name is "installInfo" may contain alphanumeric characters (at least one), and symbols, excluding double quotation marks (") and commas (,).
  Specify values up to 32 bytes.
- The setting values for lines where the section name is "parameters" may contain the following characters:

  [a-zA-Z0-9], halfwidth space, hyphen (-), backslash (\), and comma (,)
  For path names, specify up to 160 bytes.

Information

The installation parameter CSV template is "*dvdDrive*\sample\sample.csv".

1.1.2 List of Parameters

This section describes the parameters that can be set for each section.

installInfo section

The parameters that can be set in the installInfo section are shown below.

No.	Type	Parameter		Description
1	Software ID	**Parameter name**	Name	Mandatory. Specify the software identifier. For this software, specify the following value: "Fujitsu Enterprise Postgres"
		Value/Range	String	
		Default value	None	

Information

- In addition to "Name", the following parameters can be used in the "installInfo" section.
  - softwareName
  - OS
  - Version
  - Edition
- Note that specifying a parameter name other than "Name" and the parameters listed above will result in an error.
- The values set in these parameters do not affect silent installation.

Example

```
installInfo,Name,Fujitsu Enterprise Postgres
```

parameters section

The parameters that can be set in the parameters section are shown below.

Note that "<*x*>" indicates the product version.

No.	Type	Parameter		Description
1	Whether to install the server	**Parameter name**	ServerInstallExecute	Optional. Specify whether to run the installation of the server. Y: Install No: Do not install
		Value/Range	Y or N	
		Default value	Y	
2	Installation destination of the server	**Parameter name**	ServerInstallPath	Optional. Specify the installation destination of the software. The root folder (such as C:\) cannot be specified.
		Value/Range	Path name	
		Default value	C:\Program Files\Fujitsu\fsepv<*x*>server64	
3	Whether to install the client (32-bit)	**Parameter name**	Client32InstallExecute	Optional. Specify whether to run the installation of the client (32-bit).
		Value/Range	Y or N	

No.	Type	Parameter		Description
		Default value	Y	Y: Install N: Do not install
4	Installation destination of the client (32-bit)	**Parameter name**	Client32InstallPath	Optional. Specify the installation destination of the client (32-bit). The root folder (such as C:\) cannot be specified.
		Value/Range	Path name	
		Default value	C:\Program Files\Fujitsu\fsepv<*x*>client32 or C:\Program Files (x86)\Fujitsu\fsepv<*x*>client32	
5	Whether to install the client (64-bit)	**Parameter name**	Client64InstallExecute	Optional. Specify whether to run the installation of the client (64-bit). Y: Install N: Do not install This parameter is ignored on 32-bit operating systems.
		Value/Range	Y or N	
		Default value	Y	
6	Installation destination of the client (64-bit)	**Parameter name**	Client64InstallPath	Optional. Specify the installation destination of the client (64-bit). This parameter is ignored on 32-bit operating systems. The root folder (such as C:\) cannot be specified.
		Value/Range	Path name	
		Default value	C:\Program Files\Fujitsu\fsepv<*x*>client64	
7	Whether to install WebAdmin	**Parameter name**	WebAdminInstallExecute	Optional. Specify whether to run the installation of WebAdmin in multi-server mode. Y: Install N: Do not install
		Value/Range	Y or N	
		Default value	Y	
8	Installation destination of WebAdmin (64-bit)	**Parameter name**	WebAdminInstallPath	Optional. Specify the installation destination of WebAdmin. This parameter is ignored on 32-bit operating systems. The root folder (such as C:\) cannot be specified.
		Value/Range	Path name	
		Default value	C:\Program Files\Fujitsu\fsepv<*x*>webadmin64	
9	Whether to set up WebAdmin	**Parameter name**	WebSetupExecute	Optional. Specify whether to set up WebAdmin. Y: Setup N: Do not set up WebAdmin is not installed.
		Value/Range	Y or N	
		Default value	Y	
10	Web server port number	**Parameter name**	WebPortNumber1	Optional. Specify the port number of the Web server.

No.	Type	Parameter		Description
		Value/Range	1024 to 49151	WebAdmin is not installed.
		Default value	27515	
11	WebAdmin internal port number	**Parameter name**	WebPortNumber2	Optional. Specify the WebAdmin internal port number. WebAdmin is not installed.
		Value/Range	1024 to 49151	
		Default value	27516	

Example

```
installInfo,Name,Fujitsu Enterprise Postgres
parameters,ServerInstallExecute,Y
parameters,ServerInstallPath,
parameters,Client32InstallExecute,Y
parameters,Client32InstallPath,
parameters,Client64InstallExecute,Y
parameters,Client64InstallPath,
parameters,WebAdminInstallExecute,Y
parameters,WebAdminInstallPath,
parameters,WebSetupExecute,Y
parameters,WebPortNumber1,27515
parameters,WebPortNumber2,27516
```

1.1.3 Messages and Return Values

Messages are output when errors are detected during parametric analysis.

If an error is detected during installation of the product, a message is output to the log:

Log file

In "*xyz*" below, "*x*" is the product version, "*z*" is the product level, and for this release, "*y*" is 0.

```
%TEMP%\fsep_SERVER64_media_xyz_install.log
```

Note

Directories under the TEMP environment variable may be deleted when the operating system is restarted. Therefore, ensure that you browse or save the file before restarting the operating system.

Messages and return values

CSV file errors

The following messages are output if errors are detected while parsing CSV files.

Return value	Message	Explanation and actions
4	CSV file error:code = 1, Invalid CSV error.	There is an error in the specification format of the CSV file.
4	CSV file error:code = 2, installInfo/Name is required.	Either installInfo or the Name parameter has not been specified.
4	CSV file error:code = 3, Invalid installInfo key.	There is an error in the installInfo specification. Or the section name is invalid.

Return value	Message	Explanation and actions
4	CSV file error:code = 4, Duplicated installInfo key.	The same parameter has been defined more than once in installInfo.
4	CSV file error:code = 5, Invalid character length.	No setting value is specified, or the specified string is too long.
4	CSV file error:code = 6, Invalid character format or encoding.	An invalid format or character set is being used for the character string.
4	CSV file error:code = 8, Parameter is required.	There is no line in the "parameters" section.
20	The input file does not exist.	The input file does not exist.
21	The value of @1@ is incorrect,the value is @2@.	The value is incorrect. Specify the correct value. The parameter name is displayed in @1@. The specified parameter is displayed in @2@.
22	The value same at @1@ and @2@ is specified.	The same value is specified in different parameters. Specify different values. The parameter name is displayed in @1@ and @2@.
23	The @2@ of @1@ already exists.	The path already exists. Specify a different path. The parameter name is displayed in @1@. The specified value is displayed in @2@.
24	Drive @2@ of @1@ does not exist.	The drive does not exist. Specify an existing drive. The parameter name is displayed in @1@. The specified value is displayed in @2@.
26	Port number @1@ is already used in the service file.	The port number is already being used. Specify an unused port number. The port number is displayed in @1@
28	There is no product for installation.	No product was specified for installation. Specify "Y" for at least one of the following parameters: - ServerInstallExecute - Client32InstallExecute - Client64InstallExecute - WebAdminInstallExecute
29	USAGE : silent.sh inputfile	The argument specified in the command is incorrect. Specify the correct argument.

Product installer errors

The following return values are returned when errors occur while the product installer is running.

Return value	Explanation
0	Completed successfully.
3	Failed to set up WebAdmin.
14	The resources required for installation do not exist on the media.

Return value	Explanation
32	The same version of the same product is already installed.
60	An error occurred in Uninstall (middleware).
61	Failed to install Uninstall (middleware).
62	Uninstall (middleware) was started.
74	Failed to write to the registry.
79	Failed to read the registry.
83	Failed to install FJQSS or register information.
85	Different editions of the same generation are installed.
86	A newer version of the product is already installed.
88	Exclusive product is already installed.
110	License registration failed.
129	Unsupported operating system.
142	The installation destination has insufficient disk capacity.
144	The command was not executed with administrator privileges.
200	Failed to install of package.
201	A conflicting package is installed.
251	An error occurred when running a system function.
255	An internal inconsistency occurred.

1.1.4 CSV File Format

The format of CSV files is based on RFC4180, with the following specifications.

Records

- Separate each record with a "CRLF" newline (operation is not guaranteed with only a "CR" or "LF" newline).
- Specify a newline at the end of a file.
- Separate each field within a record with a comma "," (comma: ASCII (single-byte character)).

Format	Record		
aaa,bbb,ccc	aaa	bbb	ccc

- If several commas are entered in succession, or if a comma precedes a newline, the data following the comma is regarded as empty.

Format	Record		
aaa,,ccc	aaa		ccc
aaa,bbb,	aaa	bbb	

- Headers cannot be specified

Format	Record		
	field1	field2	field 3
aaa,bbb,ccc	aaa	bbb	ccc

Double quotation marks

- Enclose fields that contain newlines, double quotation marks, or commas in double quotation marks.

Format	Record		
"aaa","bb b","ccc"	aaa	bb b	ccc
"aaa","bb,b","ccc"	aaa	bb,b	ccc

- If a double quotation mark is used in a field enclosed with double quotation marks, the first double quotation mark is regarded as an escape character.

Format	Record		
"aaa","bb""b","ccc"	aaa	bb"b	ccc

Note

An error occurs in the following cases:

- Each space, tab, or whitespace character is recognized as one character, and will cause a parameter error.
- A space is entered before or after a field enclosed with double quotation marks.

```
"zzz ", "yyy ", " xxx "
```

- The number of fields differs between records.

```
aaa,bbb
aaa,bbb,ccc
aaa,bbb,ccc,ddd
```

- Fields enclosed with double quotation marks and fields not enclosed with double quotation marks are both used.

```
aaa,"bbb",ccc
"xxx",yyy,"zzz"
```

- The double quotation escape characters are not positioned correctly.

```
"aaa","bb""""b","ccc"
```

- Double quotation marks enclose the entire field.

```
"aaa,bbb,ccc"
```

1.2 Uninstallation

This chapter provides specifications for uninstallation.

1.2.1 Messages and Return Values

The uninstaller result is output to the log file.

Log file

In "*xyz*" below, "*x*" is the product version, "*z*" is the product level, and for this release, "*y*" is 0.

Server product

```
%TEMP%\fsep_SERVER64_xyz_uninstall.log
```

WebAdmin

```
%TEMP%\fsep_WEBADMIN_xyz_uninstall.log
```

Return values

The following return values are output:

Return values	Meaning
0	Uninstallation was successful.
3	WebAdmin unsetup failed.
64	Failed to install because "Uninstall (middleware)" is running.
74	Failed to write product information (updating registry).
79	Failed to read product information (viewing registry).
83	Failed to delete the information from FJQSS.
144	Not executed with administrator privileges.
200	Failed to uninstall of package.
251	An error occurred when running a system function.
255	An internal inconsistency occurred.

Chapter 2 Client Feature

This chapter provides specifications for installation and installation of the client feature in silent mode.

Note that "<*x*>" in paths indicates the product version.

2.1 Installation

This chapter provides specifications for installation.

2.1.1 Specification Format

The installation parameters CSV file, which is specified as the argument for the silent installer, has three columns per line in CSV format.

```
sectionName, parameterName, value
sectionName, parameterName, value
    :
```

Enter the following settings in respective columns.

Item	Settings	Optional
sectionName	Specify the section name. There are two types of section names: "installInfo": Set the product information. "parameters": Set the parameter information for this product.	Mandatory
parameterName	Specify the parameter name. Each section has a valid parameter.	Mandatory
value	Specify the value.	Optional

Note

- Blank lines cannot be included.
- Section names and parameter names cannot be omitted.
- Undefined parameters cannot be set in lines where the section name is "installInfo". Also, note that the same parameter cannot be specified multiple times.
- Specify at least one line with the section name "parameters".
- Undefined parameters specified in lines within the "parameters" section will be ignored during execution. Note that when the same parameter is specified multiple times, the settings in the lowest line will be valid.
- The setting values for lines where the section name is "installInfo" may contain alphanumeric characters (at least one), and symbols, excluding double quotation marks (") and commas (,).
- Do not use the following halfwidth characters in the setting values for lines with the section name "parameters".

W Windows: # $ & ' = ^ ~ | @ ` [ ] ; < > / tab

L Linux: " # $ & = ' ( ) ^ ~ \ | @ ` [ ] { } ; : < > tab

Information

The template for the installation parameters CSV file is as follows:

- W Windows client: *Z*:\sample\sample_windows.csv (*Z* is the drive into which the DVD is inserted.)

- Linux client: *mountPoint*/sample/sample_linux.csv

2.1.2 List of Parameters

This section describes the parameters that can be set for each section.

installInfo section

The parameters that can be set in the installInfo section are shown below.

No.	Type	Parameter		Description
1	Software ID	**Parameter name**	Name	Mandatory. Specify the software identifier. For this software, specify the following value: "Fujitsu Enterprise Postgres"
		Value/Range	String	
		Default value	None	

Information

- In addition to "Name", the following parameters can be used in the "installInfo" section.
  - softwareName
  - OS
  - Version
  - Edition
- Note that specifying a parameter name other than "Name" and the parameters listed above will result in an error.
- The values set in these parameters do not affect silent installation.

Example

```
installInfo,Name,Fujitsu Enterprise Postgres
```

parameters section

The parameters that can be set in the parameters section are shown below.

No.	Type	Parameter		Description
1	Whether to install the client (32-bit)	**Parameter name**	Client32InstallExecute	Optional. Specify whether to run the installation of the client (32-bit). Y: Install N: Do not install This parameter is ignored on Linux.
		Value/Range	Y or N	
		Default value	Y	
2	Installation destination of the client (32-bit)	**Parameter name**	Client32InstallPath	Optional. Specify the installation destination of the client (32-bit).
		Value/Range	Path name	

No.	Type	Parameter		Description
		Default value	W Windows(32-bit) %ProgramFiles%\Fujitsu\fsepv<*x*>client32 Windows(64-bit) %ProgramFiles(x86)%\Fujitsu\fsepv<*x*>client32 L Linux None.	W A root folder (such as C:\) cannot be specified. L This parameter is ignored on Linux.
3	Whether to install the client (64-bit)	**Parameter name**	Client64InstallExecute	Optional. Specify whether to run the installation of the client (64-bit). Y: Install N: Do not install W This parameter is ignored on 32-bit operating systems.
		Value/Range	Y or N	
		Default value	Y	
4	Installation destination of the client (64-bit)	**Parameter name**	Client64InstallPath	Optional. Specify the installation destination of the client (64-bit). W This parameter is ignored on 32-bit operating systems. A root folder (such as C:\) cannot be specified. L A root directory cannot be specified.
		Value/Range	Path name	
		Default value	W Windows(64-bit) %ProgramFiles%\Fujitsu\fsepv<*x*>client64 L Linux /opt/fsepv<*x*>client64	
5	Whether to install Pgpool-II	**Parameter name**	Pgpool2InstallExecute	Specify whether to run the installation of the Pgpool-II. Y: Install N: Do not install W This parameter is ignored on Windows.
		Value/Range	Y or N	
		Default value	N	
6	Installation destination of Pgpool-II	**Parameter name**	Pgpool2InstallPath	Optional. Specify the installation destination of Pgpool-II. W This parameter is ignored on Windows. L A root directory cannot be specified.
		Value/Range	Path name	
		Default value	/opt/fsepv<*x*>pgpool-II	

Example

```
parameters,Client64InstallExecute,Y
parameters,Client64InstallPath,
parameters,Pgpool2InstallExecute,Y
parameters,Pgpool2InstallPath,
```

2.1.3 Messages and Return Values

Messages are output when errors are detected during parametric analysis.

If an error occurs during installation of the product, a message is output to the log.

Log file

In "*xyz*" below, "*x*" is the product version, "*z*" is the product level, and for this release, "*y*" is 0.

Windows

```
%TEMP%\fsep_CLIENT_media_xyz.log
```

Linux

```
/var/log/fsep_CLIENT_media_xyz_install.log
```

Messages and return values

CSV file errors

The following messages are output if errors are detected while parsing CSV files.

Return value	Message	Explanation and actions
4	CSV file error:code = 1, Invalid CSV error.	There is an error in the specification format of the CSV file.
4	CSV file error:code = 2, installInfo/Name is required.	There is an error in the specification format of the CSV file.
4	CSV file error:code = 3, Invalid installInfo key.	There is an error in the installInfo specification. Or the section name is invalid.
4	CSV file error:code = 4, Duplicated installInfo key.	The same parameter has been defined more than once in installInfo.
4	CSV file error:code = 5, Invalid character length.	No setting value has been specified, or the specified string is too long.
4	CSV file error:code = 6, Invalid character format or encoding.	An invalid character has been specified in the installation parameters CSV file.
4	CSV file error:code = 8, Parameter is required.	The "parameters" section has no lines.
20	The input file does not exist.	The input file does not exist.
21	The value of @1@ is incorrect,the value is @2@.	The value is incorrect. Specify the correct value. The parameter name is displayed in @1@. The specified value is displayed in @2@.
22	The value same at @1@ and @2@ is specified.	The same value is specified in different parameters. Specify different values. The parameter name is displayed in @1@ and @2@.

Return value	Message	Explanation and actions
23	The @2@ of @1@ already exists.	The path already exists. Specify a different path. The parameter name is displayed in @1@. The specified value is displayed in @2@.
24	Drive @2@ of @1@ does not exist.	The drive does not exist. Specify an existing drive. The parameter name is displayed in @1@. The specified value is displayed in @2@.
28	There is no product for installation.	There is no product to be installed. Specify "Y" for at least one of the following parameters: Windows - Client32InstallExecute - Client64InstallExecute Linux - Client64InstallExecute - Pgpool2InstallExecute
29	Windows USAGE : silent.bat inputfile Linux USAGE : silent.sh inputfile	The argument specified in the command is incorrect. Specify the correct argument.

Product installer errors

The following return values are returned when errors occur while the product installer is running.

Windows

Return value	Explanation
0	Completed successfully.
14	The resources required for installation do not exist on the media.
32	The same version of the same product is already installed.
60	An error occurred in Uninstall (middleware).
61	Failed to install Uninstall (middleware).
62	Uninstall (middleware) was started.
74	Failed to write to the registry.
79	Failed to load the registry.
85	Different editions of the same generation are installed.
86	A newer version of the product is already installed.
88	Exclusive product is already installed.
129	The operating system is not supported.
142	Insufficient disk space at the installation destination.
144	The command was not executed with administrator privileges.
200	Failed to install of package.

Return value	Explanation
201	A conflicting package is installed.
220	Failed to create a folder in the installation folder.
221	Failed to create files in the installation folder.
251	An error occurred when running a system function.
255	An internal inconsistency occurred.

L Linux

Return value	Explanation
0	Completed successfully.
13	Processing was interrupted.
14	The resources required for installation do not exist on the media.
15	The resources required for installation do not exist.
16	Unable to change the permission because the directory does not exist.
17	Failed to install because a required package is missing.
19	A process is running.
30	Failed to copy the installation resources.
32	The same version of the same product is already installed.
60	An error occurred in Uninstall (middleware).
61	Failed to install Uninstall (middleware).
62	Uninstall (middleware) was started.
63	The uninstallation cannot be performed either because Uninstall (middleware) does not exist or because there are no execution privileges.
85	Different editions of the same generation are installed.
86	A newer version of the product is already installed.
88	Exclusive product is already installed.
99	A system error occurred.
106	The command was not executed with administrator privileges.
129	Unsupported operating system.
200	Failed to install of package.
201	A conflicting package is installed.

2.1.4 CSV File Format

The format of CSV files is based on RFC4180, with the following specifications.

Records

- Separate each record with a "CRLF" newline (operation is not guaranteed with only a "CR" or "LF" newline).
- Specify a newline at the end of a file.
- Separate each field within a record with a halfwidth comma ",".

Format	Record		
aaa,bbb,ccc	aaa	bbb	ccc

- If several commas are entered in succession, or if a comma precedes a newline, the data following the comma is regarded as empty.

Format	Record		
aaa,,ccc	aaa		ccc
aaa,bbb,	aaa	bbb	

- You cannot specify a header.

Format	Record		
	field1	field2	field 3
aaa,bbb,ccc	aaa	bbb	ccc

Double quotation marks

- Enclose fields that contain newlines, double quotation marks, or commas in double quotation marks.

Format	Record		
"aaa","bb b","ccc"	aaa	bb b	ccc
"aaa","bb,b","ccc"	aaa	bb,b	ccc

- If a double quotation mark is used in a field enclosed with double quotation marks, the first double quotation mark is regarded as an escape character.

Format	Record		
"aaa","bb""b","ccc"	aaa	bb"b	ccc

An error occurs in the following cases:

- Each space, tab, or whitespace character is recognized as one character, and will cause a parameter error.
- A space is entered before or after a field enclosed with double quotation marks.

```
"zzz ", "yyy ", " xxx "
```

- The number of fields differs between records.

```
aaa,bbb
aaa,bbb,ccc
aaa,bbb,ccc,ddd
```

- Fields enclosed with double quotation marks and fields not enclosed with double quotation marks are both used.

```
aaa,"bbb",ccc
"xxx",yyy,"zzz"
```

- The double quotation escape characters are not positioned correctly.

```
"aaa","bb"""b","ccc"
```

- Double quotation marks enclose the entire field.

```
"aaa,bbb,ccc"
```

2.2 Uninstallation

This chapter provides specifications for uninstallation.

2.2.1 Messages and Return Values

The uninstaller result is output to the log file.

Log file

In "*xyz*" below, "*x*" is the product version, "*z*" is the product level, and for this release, "*y*" is 0.

Windows

64-bit product

```
%TEMP%\fsep_CLIENT64_xyz_uninstall.log
```

32-bit product

```
%TEMP%\fsep_CLIENT32_xyz_uninstall.log
```

Linux

64-bit product

```
/var/log/fsep_CLIENT64_xyz_uninstall.log
```

Pgpool-II

```
/var/log/fsep_PGPOOL2_xyz_uninstall.log
```

Return values

The following return values are output:

Windows

Return values	Meaning
0	Uninstallation was successful.
64	Failed to install because "Uninstall (middleware)" is running.
74	Failed to write product information (updating registry).
79	Failed to read product information (viewing registry).
144	The command was not executed with administrator privileges.
200	Failed to uninstall of package.
251	An error occurred when running a system function.
255	An internal inconsistency occurred.

Linux

Return values	Meaning
0	Uninstallation was successful.

Return values	Meaning
13	Processing was interrupted.
15	The resources required for uninstallation do not exist.
19	A process is running.
60	An error occurred in Uninstall (middleware).
62	Uninstall (middleware) was started.
63	Uninstall (middleware) has not been installed, or there are no execution privileges.
99	A system error occurred.
100	The command argument is invalid.
106	The command was not executed with administrator privileges.
200	Failed to uninstall of package.

Chapter 3 Server Assistant

This chapter provides specifications for installation and uninstallation of the Server Assistant in silent mode.

Note that "<*x*>" in paths indicates the product version.

3.1 Installation

This chapter provides specifications for installation.

3.1.1 Specification Format

The installation parameters CSV file, which is specified as the argument for the silent installer, has three columns per line in CSV format.

```
sectionName, parameterName, value
sectionName, parameterName, value
    :
```

Enter the following settings in respective columns.

Item	Settings	Optional
sectionName	Specify the section name. There are two types of section names: "installInfo": Set the product information. "parameters": Set the parameter information for this product.	Mandatory
parameterName	Specify the parameter name. Each section has a valid parameter.	Mandatory
value	Specify the value.	Optional

Note

- Blank lines cannot be included.
- Section names and parameter names cannot be omitted.
- Undefined parameters cannot be set in lines where the section name is "installInfo". Also, note that the same parameter cannot be specified multiple times.
- Specify at least one line with the section name "parameters".
- Undefined parameters specified in lines within the "parameters" section will be ignored during execution. Note that when the same parameter is specified multiple times, the settings in the lowest line will be valid.
- The setting values for lines where the section name is "installInfo" may contain alphanumeric characters (at least one), and symbols, excluding double quotation marks (") and commas (,).
- Do not use the following halfwidth characters in the setting values for lines with the section name "parameters".

  Windows: # $ & ' = ^ ~ | @ ` [ ] ; < > / tab

  Linux: " # $ & = ' ( ) ^ ~ \ | @ ` [ ] { } ; : < > tab

Information

The template for the installation parameters CSV file is as follows:

- Windows Server Assistant: *Z*:\sample\sample.csv (*Z* is the drive into which the DVD is inserted.)

- Linux Server Assistant: *mountPoint*/sample/sample.csv

3.1.2 List of Parameters

This section describes the parameters that can be set for each section.

installInfo section

The parameters that can be set in the installInfo section are shown below.

No.	Type	Parameter		Description
1	Software ID	**Parameter name**	Name	Mandatory. Specify the software identifier. For this software, specify the following value: "Fujitsu Enterprise Postgres"
		Value/Range	String	
		Default value	None	

Information

- In addition to "Name", the following parameters can be used in the "installInfo" section.
  - softwareName
  - OS
  - Version
  - Edition
- Note that specifying a parameter name other than "Name" and the parameters listed above will result in an error.
- The values set in these parameters do not affect silent installation.

Example

```
installInfo,Name,Fujitsu Enterprise Postgres
```

parameters section

The parameters that can be set in the parameters section are shown below.

No.	Type	Parameter		Description
1	Whether to install the server assistant	**Parameter name**	ArbiterInstallExecute	Optional. Specify whether to run the installation of the Server Assistant. Y: Install N: Do not install
		Value/Range	Y or N	
		Default value	Y	
2	Installation destination of the server assistant	**Parameter name**	ArbiterInstallPath	Optional. Specify the installation destination of the Server Assistant. W A root folder (such as C:\) cannot be specified. L A root directory cannot be specified.
		Value/Range	Path name	
		Default value	W Windows %ProgramFiles%\Fujitsu\fsepv<*x*>assistant64	

No.	Type	Parameter		Description
			L Linux /opt/fsepv<*x*>assistant	

Example

```
parameters,ArbiterInstallExecute,Y
parameters,ArbiterInstallPath,
```

3.1.3 Messages and Return Values

Messages are output when errors are detected during parametric analysis.

If an error occurs during installation of the product, a message is output to the log.

Log file

In "*xyz*" below, "*x*" is the product version, "*z*" is the product level, and for this release, "*y*" is 0.

W Windows

```
%TEMP%\fsep_SERVER_ASSISTANT_media_xyz.log
```

L Linux

```
/var/log/fsep_SERVER_ASSISTANT_media_xyz_install.log
```

Messages and return values

CSV file errors

The following messages are output if errors are detected while parsing CSV files.

Return value	Message	Explanation and actions
4	CSV file error:code = 1, Invalid CSV error.	There is an error in the specification format of the CSV file.
4	CSV file error:code = 2, installInfo/Name is required.	There is an error in the specification format of the CSV file.
4	CSV file error:code = 3, Invalid installInfo key.	There is an error in the installInfo specification. Or the section name is invalid.
4	CSV file error:code = 4, Duplicated installInfo key.	The same parameter has been defined more than once in installInfo.
4	CSV file error:code = 5, Invalid character length.	No setting value has been specified, or the specified string is too long.
4	CSV file error:code = 6, Invalid character format or encoding.	An invalid character has been specified in the installation parameters CSV file.
4	CSV file error:code = 8, Parameter is required.	The "parameters" section has no lines.
20	The input file does not exist.	The input file does not exist.
21	The value of @1@ is incorrect,the value is @2@.	The value is incorrect. Specify the correct value. The parameter name is displayed in @1@. The specified value is displayed in @2@.
22	The value same at @1@ and @2@ is specified.	The same value is specified in different parameters. Specify different values.

Return value	Message	Explanation and actions
		The parameter name is displayed in @1@ and @2@.
23	The @2@ of @1@ already exists.	The path already exists. Specify a different path. The parameter name is displayed in @1@. The specified value is displayed in @2@.
24	Drive @2@ of @1@ does not exist.	The drive does not exist. Specify an existing drive. The parameter name is displayed in @1@. The specified value is displayed in @2@.
28	There is no product for installation.	There is no product to be installed. Specify "Y" for the following parameters: - ArbiterInstallExecute
29	Windows USAGE : silent.bat inputfile Linux USAGE : silent.sh inputfile	The argument specified in the command is incorrect. Specify the correct argument.

Product installer errors

The following return values are returned when errors occur while the product installer is running.

Windows

Return value	Explanation
0	Completed successfully.
14	The resources required for installation do not exist on the media.
32	The same version of the same product is already installed.
60	An error occurred in Uninstall (middleware).
61	Failed to install Uninstall (middleware).
62	Uninstall (middleware) was started.
74	Failed to write to the registry.
79	Failed to load the registry.
83	Failed to install FJQSS or register information.
86	A newer version of the product is already installed.
88	Exclusive product is already installed.
129	The operating system is not supported.
142	Insufficient disk space at the installation destination.
144	The command was not executed with administrator privileges.
200	Failed to install of package.
201	A conflicting package is installed.
220	Failed to create a folder in the installation folder.
221	Failed to create files in the installation folder.
251	An error occurred when running a system function.

Return value	Explanation
255	An internal inconsistency occurred.

Linux

Return value	Explanation
0	Completed successfully.
13	Processing was interrupted.
14	The resources required for installation do not exist on the media.
15	The resources required for installation do not exist.
16	Unable to change the permission because the directory does not exist.
17	Failed to install because a required package is missing.
19	A process is running.
30	Failed to copy the installation resources.
32	The same version of the same product is already installed.
60	An error occurred in Uninstall (middleware).
61	Failed to install Uninstall (middleware).
62	Uninstall (middleware) was started.
63	The uninstallation cannot be performed either because Uninstall (middleware) does not exist or because there are no execution privileges.
83	Failed to install FJQSS or register information.
86	A newer version of the product is already installed.
88	Exclusive product is already installed.
99	A system error occurred.
106	The command was not executed with administrator privileges.
129	Unsupported operating system.
200	Failed to install of package.
201	A conflicting package is installed.

3.1.4 CSV File Format

The format of CSV files is based on RFC4180, with the following specifications.

Records

- Separate each record with a "CRLF" newline (operation is not guaranteed with only a "CR" or "LF" newline).
- Specify a newline at the end of a file.
- Separate each field within a record with a halfwidth comma ",".

Format	Record		
aaa,bbb,ccc	aaa	bbb	ccc

- If several commas are entered in succession, or if a comma precedes a newline, the data following the comma is regarded as empty.

Format	Record		
aaa,,ccc	aaa		ccc
aaa,bbb,	aaa	bbb	

- You cannot specify a header.

Format	Record		
	field1	field2	field 3
aaa,bbb,ccc	aaa	bbb	ccc

Double quotation marks

- Enclose fields that contain newlines, double quotation marks, or commas in double quotation marks.

Format	Record		
"aaa","bb b","ccc"	aaa	bb b	ccc
"aaa","bb,b","ccc"	aaa	bb,b	ccc

- If a double quotation mark is used in a field enclosed with double quotation marks, the first double quotation mark is regarded as an escape character.

Format	Record		
"aaa","bb""b","ccc"	aaa	bb"b	ccc

Note

An error occurs in the following cases:

- Each space, tab, or whitespace character is recognized as one character, and will cause a parameter error.
- A space is entered before or after a field enclosed with double quotation marks.

```
"zzz ", "yyy ", " xxx "
```

- The number of fields differs between records.

```
aaa,bbb
aaa,bbb,ccc
aaa,bbb,ccc,ddd
```

- Fields enclosed with double quotation marks and fields not enclosed with double quotation marks are both used.

```
aaa,"bbb",ccc
"xxx",yyy,"zzz"
```

- The double quotation escape characters are not positioned correctly.

```
"aaa","bb"""b","ccc"
```

- Double quotation marks enclose the entire field.

```
"aaa,bbb,ccc"
```

3.2 Uninstallation

This chapter provides specifications for uninstallation.

3.2.1 Messages and Return Values

The uninstaller result is output to the log file.

Log file

In "*xyz*" below, "*x*" is the product version, "*z*" is the product level, and for this release, "*y*" is 0.

Windows

```
%TEMP%\fsep_SERVER_ASSISTANT_xyz_uninstall.log
```

Linux

```
/var/log/fsep_SERVER_ASSISTANT_xyz_uninstall.log
```

Return values

The following return values are output:

Windows

Return values	Meaning
0	Uninstallation was successful.
64	Failed to install because "Uninstall (middleware)" is running.
74	Failed to write product information (updating registry).
79	Failed to read product information (viewing registry).
83	Failed to delete the information from FJQSS.
144	The command was not executed with administrator privileges.
200	Failed to uninstall of package.
251	An error occurred when running a system function.
255	An internal inconsistency occurred.

Linux

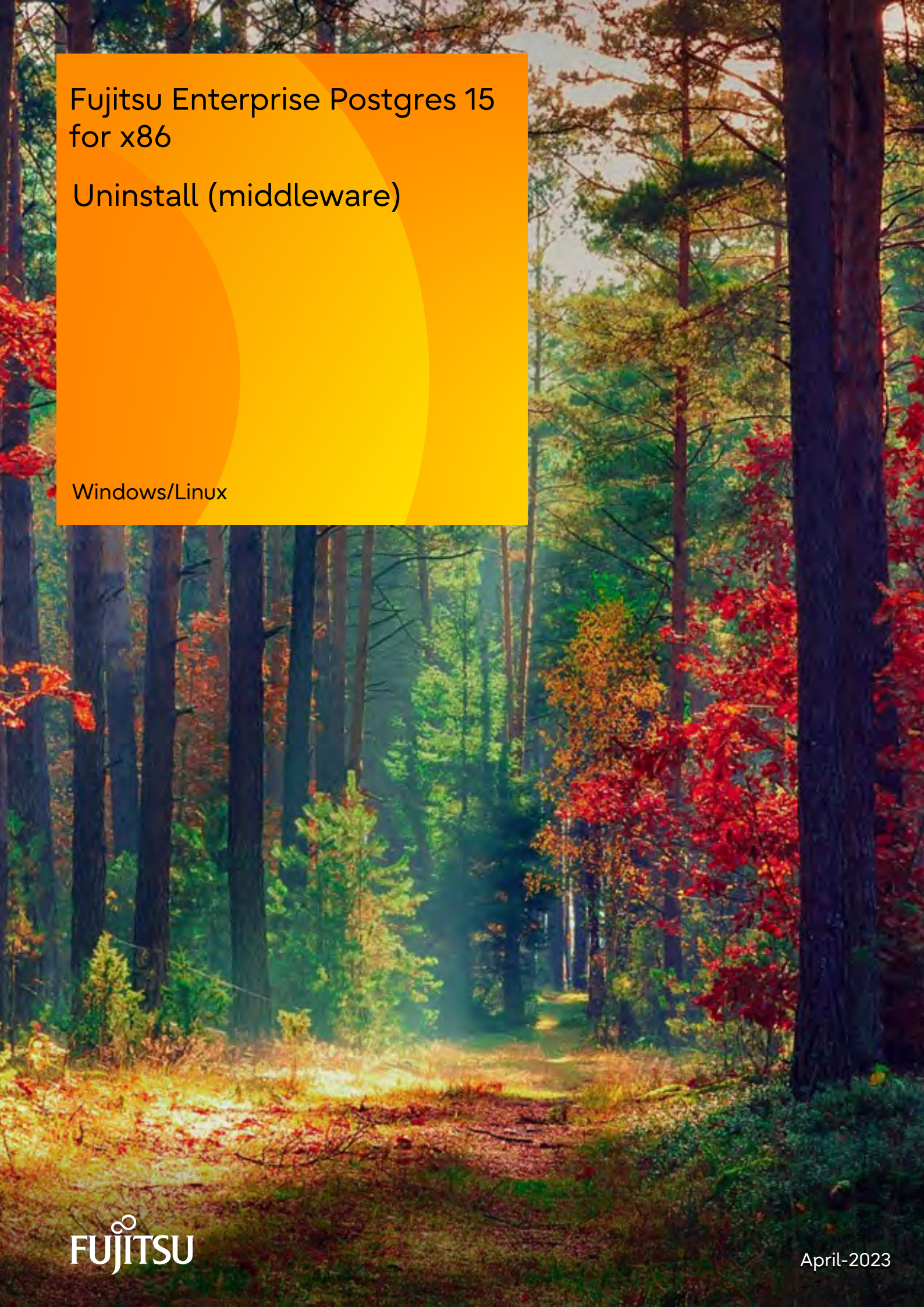
Return values	Meaning
0	Uninstallation was successful.
13	Processing was interrupted.
15	The resources required for uninstallation do not exist.
19	A process is running.
60	An error occurred in Uninstall (middleware).
62	Uninstall (middleware) was started.
63	Uninstall (middleware) has not been installed, or there are no execution privileges.
83	Failed to delete the information from FJQSS.
99	A system error occurred.
100	The command argument is invalid.

Return values	Meaning
106	The command was not executed with administrator privileges.
200	Failed to uninstall of package.

Fujitsu Enterprise Postgres 15 for x86

Uninstall (middleware)

Windows/Linux



FUJITSU

April-2023

Preface

Purpose of this document

This document explains the features, notes, and messages of Uninstall (middleware).

Structure of this document

This document is structured as follows:

Chapter 1 Using Uninstall (middleware)

Describes the features and notes of Uninstall (middleware).

Chapter 2 Uninstall (middleware) Messages

Describes the messages output by Uninstall (middleware).

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version


```
Edition 2.0: April 2023
Edition 1.0: April 2021
```


Copyright

Copyright 2018-2023 Fujitsu Limited

Contents

Chapter 1 Using Uninstall (middleware)....1
1.1 Features that are Installed....1
1.2 Notes....1
1.2.1 Notes on the Uninstall (middleware) Tool....1

Chapter 2 Uninstall (middleware) Messages....3
2.1 Messages output by FJSVcir....3
2.2 Messages output by Uninstall (middleware)....6

Chapter 1 Using Uninstall (middleware)

1.1 Features that are Installed

[Windows]

Feature	Package name	Component name	Remarks	Selectively installed?
Common tool	FJSVcir	CIRuntime Application	Controls the installation and uninstallation of Fujitsu middleware products. Manages the installation information of Fujitsu middleware products and includes a management function for viewing installation information and launching the uninstaller of each product. ("Uninstall (middleware)")	No

[Linux/Solaris]

Feature	Package name	Component name	Remarks	Selectively installed?
Common tool	FJSVcir	CIRuntime Application	Controls the installation and uninstallation of Fujitsu middleware products. Manages the installation information of Fujitsu middleware products and includes a management function for viewing installation information and launching the uninstaller of each product. ("Uninstall (middleware)")	No

1.2 Notes

1.2.1 Notes on the Uninstall (middleware) Tool

"Uninstall (middleware)" is a tool used by Fujitsu middleware products. It performs information management for middleware products, and also, launching the uninstaller for removing these products from the installed systems.

Note

- To uninstall Fujitsu Enterprise Postgres, uninstall from "Uninstall (middleware)".
- This tool manages information about other Fujitsu middleware products as well as Fujitsu Enterprise Postgres. For this reason, do not uninstall it unless absolutely necessary. If you have accidentally uninstalled this tool, re-install it as described below.

  [Windows]

  1. Log on to the machine to be installed using a user name belonging to the Administrators group, or switch to an account with administrator privileges.
  2. Insert the server program DVD into the drive device.
  3. Execute the installation command.

     ```
     z:\CIR\Windows\cirinst.exe
     ```

     In the example above, *z* is the drive into which the DVD is inserted.

[Linux/Solaris]

1. Log in as a superuser to the installation target machine, or switch to a user with administrative privileges.
2. Insert the product media in the drive.
3. Execute the installation command.

   [Linux]

   ```
   # /media/dvd/CIR/Linux/cirinst.sh
   ```

   In the example above, /media/dvd is the DVD mount point.

   [Solaris]

   ```
   # /dvd/dvd0/CIR/Solaris/cirinst.sh
   ```

   In the example above, /dvd/dvd0 is the DVD mount point.

- If you uninstall this tool as described below.

1. Check that there are no Fujitsu middleware products installed in the system.
   You can check this by using "Uninstall (middleware)", which is launched by the command below.

   [Windows]

   Select [All Programs] > [Fujitsu] > [Uninstall (middleware)]

   [Linux/Solaris]

   ```
   /opt/FJSVcir/cir/bin/cimanager.sh -c
   ```

   -c: CUI

2. If no Fujitsu middleware products have been installed, then issue the command below.

   [Windows]

   ```
   %SystemDrive%\FujitsuF4CR\bin\cirremove.exe
   ```

   [Linux/Solaris]

   ```
   # /opt/FJSVcir/bin/cirremove.sh
   ```

3. At the uninstallation confirmation prompt, enter "y".

   ```
   This software is a common tool of Fujitsu products.Are you sure you want to remove it?[y/n]:
   ```

   Uninstallation will start.

Chapter 2 Uninstall (middleware) Messages

2.1 Messages output by FJSVcir

FSP_FJSVCIR_CIRINST: ERROR: 101: CIRINST101: Administrator privilege is required.

Description

The user does not have administrator privileges.

Action method

Login with administrator privileges and run the command again.

FSP_FJSVCIR_CIRINST: ERROR: 102: CIRINST102: Failed to create *%s*

Parameters

%s: Directory name

Description

Failed to create a directory during the installation.

Action method

Creating the directory might have failed for some reason. Run the installer again, or run cirinst command again. If the problem persists, check the system log.

FSP_FJSVCIR_CIRINST: ERROR: 103: CIRINST103: It is required at least *%d*MB

Parameters

%d: Disk spaces

Description

Insufficient disk space.

Action method

Confirm that the system has sufficient disk spaces.

[Windows] FSP_FJSVCIR_CIRINST: ERROR: 104: CIRINST104: Failed to copy CIR files.

[Linux][Solaris] FSP_FJSVCIR_CIRINST: ERROR: 104: CIRINST104: Failed to copy CIR installer files.

Description

Failed to copy files required for the installation.

Action method

Copying the files failed for some reason. Run the installer again, or run cirinst command again. If the problem persists, check the system log.

FSP_FJSVCIR_CIRINST: ERROR: 105: CIRINST105: Failed to copy JRE.

Description

Failed to copy JRE required for installation.

Action method

Copying the files failed for some reason. Run the installer again, or run cirinst command again. If the problem persists, check the system log.

FSP_FJSVCIR_CIRINST: ERROR: 106: CIRINST106: Failed to copy CIR

Description

Failed to copy files during the installation.

Action method

Copying the files failed for some reason. Run the installer again, or run cirinst command again. If the problem persists, check the system log.

FSP_FJSVCIR_CIRINST: ERROR: 107: CIRINST107: invalid option.

Description

An invalid option was specified.

Action method

Specify a valid option. [-c]: CUI

FSP_FJSVCIR_CIRINST: ERROR: 108: CIRINST108: JRE which should be deployed is not bundled.

Description

An appropriate JRE for this system is not bundled.

Action method

Confirm that the product supports the platform.

FSP_FJSVCIR_CIRINST: ERROR: 109: CIRINST109: Unsupported OS.

Description

Unsupported operating system.

Action method

Confirm that the product supports the operating system.

FSP_FJSVCIR_CIRINST: ERROR: 185: CIRINST185: lsb_release was not found. Please install the package below.
- redhat-lsb

Description

lsb_release command was not found.

Action method

Add redhat-lsb package and then execute install again.

FSP_FJSVCIR_CIRINST: ERROR: 186: CIRINST186: lsb_release was not found.

Description

lsb_release command was not found.

Action method

Add the relevant package and then execute install again.

FSP_FJSVCIR_CIRINST: ERROR: 201: CIRINST201: Administrator privilege is required.

Description

The user does not have administrator privileges.

Action method

Login with administrator privileges and run the command again.

FSP_FJSVCIR_CIRINST: ERROR: 203: CIRINST203: Failed to delete files.

Description

Failed to delete FJSVcir files.

Action method

Deleting the files failed for some reason. Run cirremove command again. If the problem persists, check the system log.

FSP_FJSVCIR_CIRINST: ERROR: 205: CIRINST205: invalid option.

Description

Invalid option is specified.

Action method

No option is supported. Retry without any options.

Specified installation parameters file was not found.

Description

Specified installation parameters file was not found. (silent mode)

Action method

Specify the installation parameters file.

It failed to install.See log for details

Description

An error occurred during install. (silent mode)

Action method

Collect the files stored in the following directories.

[UNIX]

/var/opt/FJSVcir/cir/logs/cirlog0.log.0

[Windows]

%ProgramData%\Fujitsu\FujitsuF4CR\cir\logs\cirlog0.log.0

Parameters are invalid.Please specifiy valid parameters.

Description

Invalid option is specified. (silent mode)

Action method

Specify a valid option.

Internal data is broken.

Description

Internal data used by this product is corrupt.

Action method

Collect the files stored in the following directories and contact Fujitsu technical support:

[UNIX]

/var/opt/FJSVCIR/cir/

/etc/opt/FJSVCIR/cir/CIR.properties

[Windows]

%ProgramData%\Fujitsu\FujitsuF4CR\cir\

2.2 Messages output by Uninstall (middleware)

It has already been running in another process.

Description

An attempt was made to start Uninstall (middleware) more than once. Alternatively, an attempt was made to start Uninstall (middleware) while the installer was starting.

Action method

End any Uninstall (middleware) that have already started. Alternatively, start Uninstall (middleware) after the installer has ended.

Specified option is invalid.

Description

An invalid option was specified.

Action method

Specify valid options and run the command again.

Please enter a valid option.

Description

An invalid character was entered. Valid characters are y, n, or item numbers.

Action method

Enter a valid character.

Failed to uninstall of the following software packages:

Description

Failed to uninstall the software listed in this message.

Action method

Confirm the information shown after this message and take the appropriate action.

Unable to uninstall the software as it is referred from more than one software.

Description

Failed to uninstall the software because it is used by other software.

Action method

No specific action is required.

Unable to uninstall the software as the software is mandatory.

Description

Failed to uninstall the software because it is required by another program.

Action method

No specific action is required.

Internal data is broken.

Description

Internal data used by this product is corrupt.

Action method

Collect the files stored in the following directories and contact Fujitsu technical support:

[UNIX]

/var/opt/FJSVCIR/cir/

/etc/opt/FJSVCIR/cir/CIR.properties

[Windows]

%ProgramData%\Fujitsu\FujitsuF4CR\cir\

Unable to uninstall the software you were about to uninstall as the existence of the software is a precondition for the operation of another software.

Description

Failed to uninstall the software because it is required by another program.

Action method

No specific action is required.

The program terminated abnormally.

Description

The program has terminated abnormally.

Action method

Collect the files stored in the following directories and contact Fujitsu technical support:

[UNIX]

/var/opt/FJSVCIR/cir/

/etc/opt/FJSVCIR/cir/CIR.properties

[Windows]

%ProgramData%\Fujitsu\FujitsuF4CR\cir\

An unexpected error has occurred during uninstall.

Description

An error occurred during uninstall.

Action method

Collect the files stored in the following directories and contact Fujitsu technical support:

[UNIX]

/var/opt/FJSVCIR/cir/

/etc/opt/FJSVCIR/cir/CIR.properties

[Windows]

%ProgramData%\Fujitsu\FujitsuF4CR\cir\

It failed to uninstall.See log for details.

Description

An error occurred during uninstall. (silent mode)

Action method

Collect the files stored in the following directories.

[UNIX]

/var/opt/FJSVcir/cir/logs/cirlog0.log.0

[Windows]

%ProgramData%\Fujitsu\FujitsuF4CR\cir\logs\cirlog0.log.0

Failed to initialize the temp directory.

Description

Can not start Uninstall (middleware) because failed to initialize the temp directory.

Action method

Run Uninstall (middleware) again. If the problem persists, check whether other processes have accessed the files in the following directories.

[UNIX]

/var/opt/FJSVcir/cir/temp/meta_db

[Windows]

%ProgramData%\Fujitsu\FujitsuF4CR\cir\temp\meta_db

[Notice]Need to restart for uninstall completion.

Description

Uninstallation was completed. (silent mode)

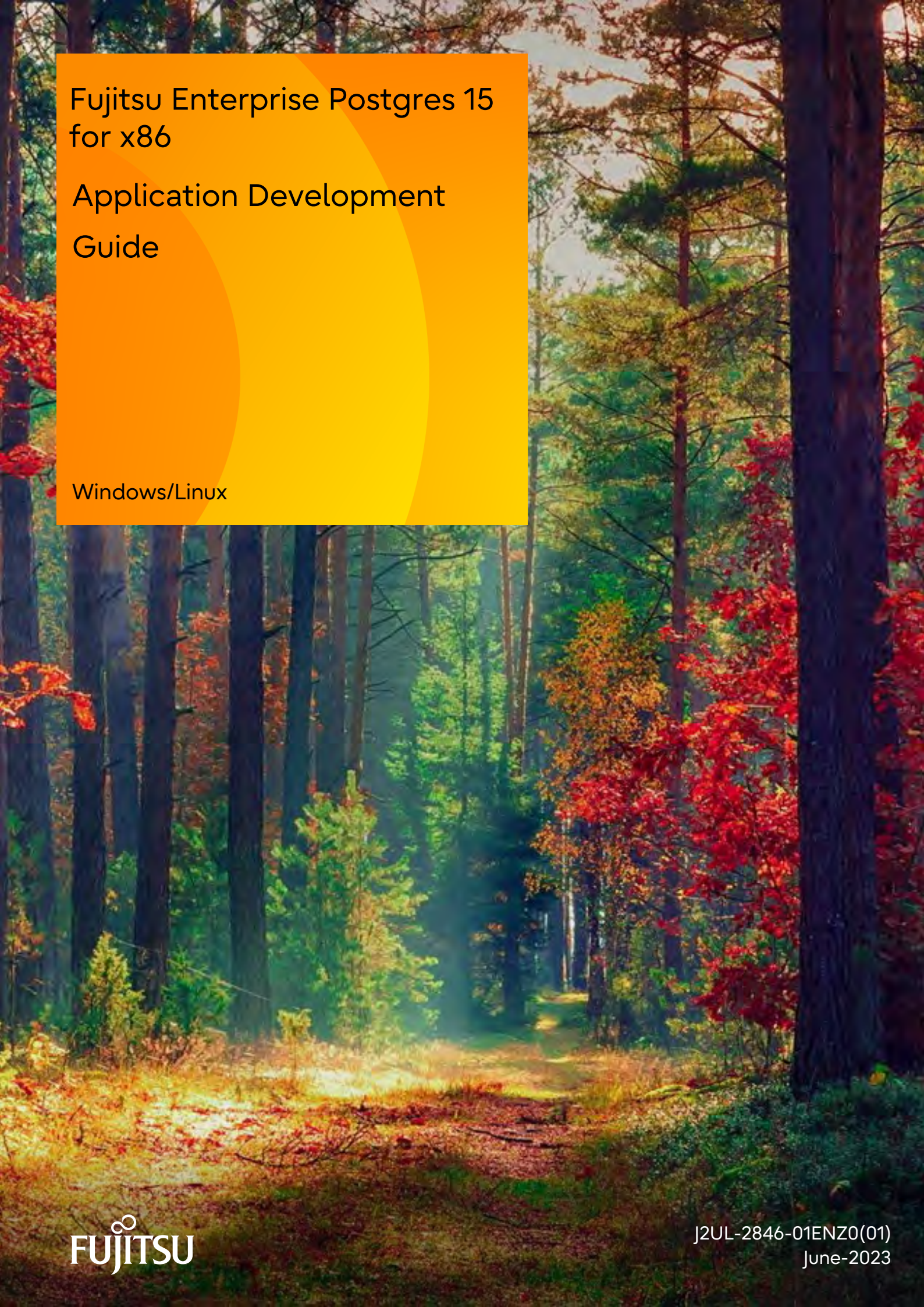
Action method

Restart the system.

Fujitsu Enterprise Postgres 15 for x86

Application Development Guide

Windows/Linux



FUJITSU

J2UL-2846-01ENZ0(01)
June-2023

Preface

Purpose of this document

This is a guide for the developers of Fujitsu Enterprise Postgres applications.

Intended readers

This document is intended for developers of applications that use Fujitsu Enterprise Postgres. Of the interfaces provided by Fujitsu Enterprise Postgres, this guide describes the PostgreSQL extended interface.

Readers of this document are also assumed to have general knowledge of:

L
- PostgreSQL
- SQL
- Linux

W
- PostgreSQL
- SQL
- Windows

Structure of this document

This document is structured as follows:

Chapter 1 Overview of the Application Development Function

Provides an overview of Fujitsu Enterprise Postgres application development.

Chapter 2 JDBC Driver

Explains how to use JDBC drivers.

Chapter 3 ODBC Driver

Explains how to use ODBC drivers.

Chapter 4 .NET Data Provider

Explains how to use .NET Data Provider.

Chapter 5 C Library (libpq)

Explains how to use C applications.

Chapter 6 Embedded SQL in C

Explains how to use embedded SQL in C.

Chapter 7 Embedded SQL in COBOL

Explains how to use embedded SQL in COBOL.

Chapter 8 SQL References

Explains the SQL statements which were extended in Fujitsu Enterprise Postgres development.

Chapter 9 Compatibility with Oracle Databases

Explains features that are compatible with Oracle databases.

Chapter 10 Application Connection Switch Feature

Explains the application connection switch feature.

Chapter 11 Performance Tuning

Explains how to tune application performance.

Chapter 12 Scan Using a Vertical Clustered Index (VCI)

Explains how to perform scan using a Vertical Clustered Index (VCI).

Appendix A Precautions when Developing Applications

Provides some points to note about application development.

Appendix B Conversion Procedures Required due to Differences from Oracle Database

Explains how to convert from an Oracle database to Fujitsu Enterprise Postgres, within the scope noted in "Compatibility with Oracle Databases" from the following perspectives.

Appendix C Tables Used by the Features Compatible with Oracle Databases

Explains the tables used by the features compatible with Oracle databases.

Appendix D ECOBPG - Embedded SQL in COBOL

Explains application development using embedded SQL in COBOL.

Appendix E Quantitative Limits

This appendix explains limitations.

Appendix F Reference

Provides a reference for each interface.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

```
Edition 1.1: June  2023
Edition 1.0: April  2023
```

Copyright

Copyright 2015-2023 Fujitsu Limited

Revision History

Revision	Location	Version
Fix misnamed NuGet package	4.2.3 Setting Up Entity Framework Core	Edition1.1

Contents

Chapter 1 Overview of the Application Development Function 1
1.1 Support for National Characters 2
1.1.1 Literal 2
1.1.2 Data Type 2
1.1.3 Functions and Operator 3
1.2 Compatibility with Oracle Database 3
1.3 Application Connection Switch Feature 4
1.3.1 Integration with Database Multiplexing 4
1.4 Notes on Application Compatibility 4
1.4.1 Checking Execution Results 4
1.4.2 Referencing System Catalogs 4
1.4.3 Using Functions 5

Chapter 2 JDBC Driver 6
2.1 Development Environment 6
2.1.1 Combining with JDK or JRE 6
2.2 Setup 6
2.2.1 Environment Settings 6
2.2.2 Message Language and Encoding System Used by Applications Settings 7
2.2.3 Settings for Encrypting Communication Data 7
2.3 Connecting to the Database 8
2.3.1 Using the DriverManager Class 8
2.3.2 Using the PGConnectionPoolDataSource Class 9
2.3.3 Using the PGXADataSource Class 10
2.4 Application Development 11
2.4.1 Relationship between the Application Data Types and Database Data Types 11
2.4.2 Statement Caching Feature 12
2.4.3 Creating Applications while in Database Multiplexing Mode 12
2.4.3.1 Errors when an Application Connection Switch Occurs and Corresponding Actions 13

Chapter 3 ODBC Driver 14
3.1 Development Environment 14
3.2 Setup 14
3.2.1 Registering ODBC Drivers 14
3.2.2 Registering ODBC Data Sources(for Windows(R)) 16
3.2.2.1 Registering Using GUI 16
3.2.2.2 Registering Using Commands 18
3.2.3 Registering ODBC Data Sources(for Linux) 21
3.2.4 Message Language and Encoding System Used by Applications Settings 23
3.3 Connecting to the Database 24
3.4 Application Development 24
3.4.1 Compiling Applications (for Windows (R)) 24
3.4.2 Compiling Applications (for Linux) 25
3.4.3 Creating Applications While in Database Multiplexing Mode 25
3.4.3.1 Errors when an Application Connection Switch Occurs and Corresponding Actions 26

Chapter 4 .NET Data Provider 27
4.1 Development Environment 27
4.2 Setup 27
4.2.1 Setting Up .NET Data Provider 27
4.2.2 Setting Up .NET Data Provider Type Plugins 28
4.2.3 Setting Up Entity Framework Core 28
4.2.4 Setting Up Npgsql.OpenTelemetry 29
4.2.5 Message Language Settings 29
4.3 Connecting to the Database 30
4.3.1 Using NpgsqlConnection 30

4.3.2 Using NpgsqlConnectionStringBuilder ... 30
4.3.3 Connection String ... 30
4.4 Application Development ... 35
4.4.1 Data Types ... 35
4.4.2 Relationship between Application Data Types and Database Data Types ... 37
4.4.3 Creating Applications while in Database Multiplexing Mode ... 40
4.4.3.1 Errors when an Application Connection Switch Occurs and Corresponding Actions ... 40
4.4.4 Notes ... 41
4.5 Uninstallation ... 45
4.5.1 Uninstalling Npgsql ... 45
4.5.2 Uninstalling .NET Data Provider Type Plugins ... 46
4.5.3 Uninstalling Entity Framework Core ... 46
4.5.4 Uninstalling Npgsql.OpenTelemetry ... 46

Chapter 5 C Library (libpq) ... 47
5.1 Development Environment ... 47
5.2 Setup ... 47
5.2.1 Environment Settings ... 47
5.2.2 Message Language and Encoding System Used by Applications Settings ... 48
5.2.3 Settings for Encrypting Communication Data ... 49
5.3 Connecting with the Database ... 49
5.4 Application Development ... 49
5.4.1 Compiling Applications ... 50
5.4.2 Creating Applications while in Database Multiplexing Mode ... 50
5.4.2.1 Errors when an Application Connection Switch Occurs and Corresponding Actions ... 51

Chapter 6 Embedded SQL in C ... 52
6.1 Development Environment ... 52
6.2 Setup ... 52
6.2.1 Environment Settings ... 52
6.2.2 Message Language and Encoding System Used by Applications Settings ... 52
6.2.3 Settings for Encrypting Communication Data ... 52
6.3 Connecting with the Database ... 53
6.4 Application Development ... 55
6.4.1 Support for National Character Data Types ... 55
6.4.2 Compiling Applications ... 55
6.4.3 Bulk INSERT ... 57
6.4.4 Creating Applications while in Database Multiplexing Mode ... 61
6.4.4.1 Errors when an Application Connection Switch Occurs and Corresponding Actions ... 62
6.4.5 Notes ... 62

Chapter 7 Embedded SQL in COBOL ... 63
7.1 Development Environment ... 63
7.2 Setup ... 63
7.2.1 Environment Settings ... 63
7.2.2 Message Language and Encoding System Used by Applications ... 63
7.2.3 Settings for Encrypting Communication Data ... 63
7.3 Connecting with the Database ... 63
7.4 Application Development ... 65
7.4.1 Support for National Character Data Types ... 65
7.4.2 Compiling Applications ... 67
7.4.3 Bulk INSERT ... 69
7.4.4 DECLARE STATEMENT ... 73
7.4.5 Creating Applications while in Database Multiplexing Mode ... 73
7.4.5.1 Errors when an Application Connection Switch Occurs and Corresponding Actions ... 74

Chapter 8 SQL References ... 75
8.1 Expanded Trigger Definition Feature ... 75

8.1.1 CREATE TRIGGER....75
8.1.2 How to Define Triggers in pgAdmin....76

Chapter 9 Compatibility with Oracle Databases....77
9.1 Overview....77
9.2 Precautions when Using the Features Compatible with Oracle Databases....77
9.2.1 Notes on search_path....78
9.2.2 Notes on SUBSTR....78
9.2.3 Notes when Integrating with the Interface for Application Development....78
9.3 Queries....78
9.3.1 Outer Join Operator (+)....78
9.3.2 DUAL Table....81
9.4 SQL Function Reference....81
9.4.1 DECODE....81
9.4.2 SUBSTR....83
9.4.3 NVL....84
9.5 Package Reference....85
9.5.1 DBMS_SQL....86
9.5.1.1 Description....87
9.5.1.2 Example....91

Chapter 10 Application Connection Switch Feature....94
10.1 Connection Information for the Application Connection Switch Feature....94
10.2 Using the Application Connection Switch Feature....95
10.2.1 Using the JDBC Driver....95
10.2.2 Using the ODBC Driver....97
10.2.3 Using a .NET Data Provider....99
10.2.4 Using a Connection Service File....100
10.2.5 Using the C Library (libpq)....102
10.2.6 Using Embedded SQL....103
10.2.7 Using the psql Command....105

Chapter 11 Performance Tuning....107
11.1 Enhanced Query Plan Stability....107
11.1.1 Optimizer Hints....107
11.1.2 Locked Statistics....109

Chapter 12 Scan Using a Vertical Clustered Index (VCI)....113
12.1 Operating Conditions....113
12.2 Usage....114
12.2.1 Designing....115
12.2.2 Checking....116
12.2.3 Evaluating....117
12.3 Usage Notes....117

Appendix A Precautions when Developing Applications....119
A.1 Precautions when Using Functions and Operators....119
A.1.1 General rules of Functions and Operators....119
A.1.2 Errors when Developing Applications that Use Functions and/or Operators....119
A.2 Notes when Using Temporary Tables....120
A.3 Implicit Data Type Conversions....120
A.3.1 Function Argument....122
A.3.2 Operators....122
A.3.3 Storing Values....123
A.4 Notes on Using Index....123
A.4.1 SP-GiST Index....123
A.5 Notes on Using Multibyte Characters in Definition Names....123
A.6 How to Build and Run an Application that Uses Shared Libraries....124

A.6.1 Setting DT_RUNPATH for Applications....125
A.6.2 Direct Linking of Indirectly Used Libraries to Applications....126

Appendix B Conversion Procedures Required due to Differences from Oracle Database....129
B.1 Outer Join Operator (Perform Outer Join)....129
B.1.1 Comparing with the ^= Comparison Operator....129
B.2 DECODE (Compare Values and Return Corresponding Results)....130
B.2.1 Comparing Numeric Data of Character String Types and Numeric Characters....130
B.2.2 Obtaining Comparison Result from more than 50 Conditional Expressions....130
B.2.3 Obtaining Comparison Result from Values with Different Data Types....131
B.3 SUBSTR (Extract a String of the Specified Length from Another String)....132
B.3.1 Specifying a Value Expression with a Data Type Different from the One that can be Specified for Function Arguments....132
B.3.2 Extracting a String with the Specified Format from a Datetime Type Value....133
B.3.3 Concatenating a String Value with a NULL value....133
B.4 NVL (Replace NULL)....134
B.4.1 Obtaining Result from Arguments with Different Data Types....134
B.4.2 Operating on Datetime/Numeric, Including Adding Number of Days to a Particular Day....135
B.4.3 Calculating INTERVAL Values, Including Adding Periods to a Date....135
B.5 DBMS_OUTPUT (Output Messages)....136
B.5.1 Outputting Messages Such As Process Progress Status....136
B.5.2 Receiving a Return Value from a Procedure (PL/SQL) Block (For GET_LINES)....138
B.5.3 Receiving a Return Value from a Procedure (PL/SQL) Block (For GET_LINE)....140
B.6 UTL_FILE (Perform File Operation)....141
B.6.1 Registering a Directory to Load and Write Text Files....141
B.6.2 Checking File Information....142
B.6.3 Copying Files....146
B.6.4 Moving/Renaming Files....147
B.7 DBMS_SQL (Execute Dynamic SQL)....148
B.7.1 Searching Using a Cursor....148

Appendix C Tables Used by the Features Compatible with Oracle Databases....154
C.1 UTL_FILE.UTL_FILE_DIR....154

Appendix D ECOBPG - Embedded SQL in COBOL....155
D.1 Precautions when Using Functions and Operators....155
D.2 Managing Database Connections....156
D.2.1 Connecting to the Database Server....156
D.2.2 Choosing a Connection....157
D.2.3 Closing a Connection....158
D.3 Running SQL Commands....158
D.3.1 Executing SQL Statements....158
D.3.2 Using Cursors....159
D.3.3 Managing Transactions....159
D.3.4 Prepared Statements....159
D.4 Using Host Variables....160
D.4.1 Overview....160
D.4.2 Declare Sections....160
D.4.3 Retrieving Query Results....161
D.4.4 Type Mapping....162
D.4.5 Handling Nonprimitive SQL Data Types....166
D.4.6 Indicators....170
D.5 Dynamic SQL....171
D.5.1 Executing Statements without a Result Set....171
D.5.2 Executing a Statement with Input Parameters....171
D.5.3 Executing a Statement with a Result Set....171
D.6 Using Descriptor Areas....172
D.6.1 Named SQL Descriptor Areas....172
D.7 Error Handling....174

D.7.1 Setting Callbacks ... 174
D.7.2 sqlca ... 176
D.7.3 SQLSTATE vs. SQLCODE ... 177
D.8 Preprocessor Directives ... 180
D.8.1 Including Files ... 180
D.8.2 The define and undef Directives ... 180
D.8.3 ifdef, ifndef, else, elif, and endif Directives ... 181
D.9 Processing Embedded SQL Programs ... 181
D.10 Large Objects ... 182
D.11 Embedded SQL Commands ... 182
D.11.1 ALLOCATE DESCRIPTOR ... 182
D.11.2 CONNECT ... 183
D.11.3 DEALLOCATE DESCRIPTOR ... 185
D.11.4 DECLARE ... 186
D.11.5 DESCRIBE ... 187
D.11.6 DISCONNECT ... 187
D.11.7 EXECUTE IMMEDIATE ... 188
D.11.8 GET DESCRIPTOR ... 189
D.11.9 OPEN ... 191
D.11.10 PREPARE ... 191
D.11.11 SET AUTOCOMMIT ... 192
D.11.12 SET CONNECTION ... 193
D.11.13 SET DESCRIPTOR ... 193
D.11.14 TYPE ... 194
D.11.15 VAR ... 195
D.11.16 WHENEVER ... 196
D.12 PostgreSQL Client Applications ... 197
D.12.1 ecobpg ... 197

Appendix E Quantitative Limits ... 200

Appendix F Reference ... 205
F.1 JDBC Driver ... 205
F.2 ODBC Driver ... 205
F.2.1 List of Supported APIs ... 205
F.3 .NET Data Provider ... 208
F.4 C Library (libpq) ... 208
F.5 Embedded SQL in C ... 208

Index ... 209

Chapter 1 Overview of the Application Development Function

The interface for application development provided by Fujitsu Enterprise Postgres is perfectly compatible with PostgreSQL.

Along with the PostgreSQL interface, Fujitsu Enterprise Postgres also provides the following extended interfaces:

- Support for National Characters

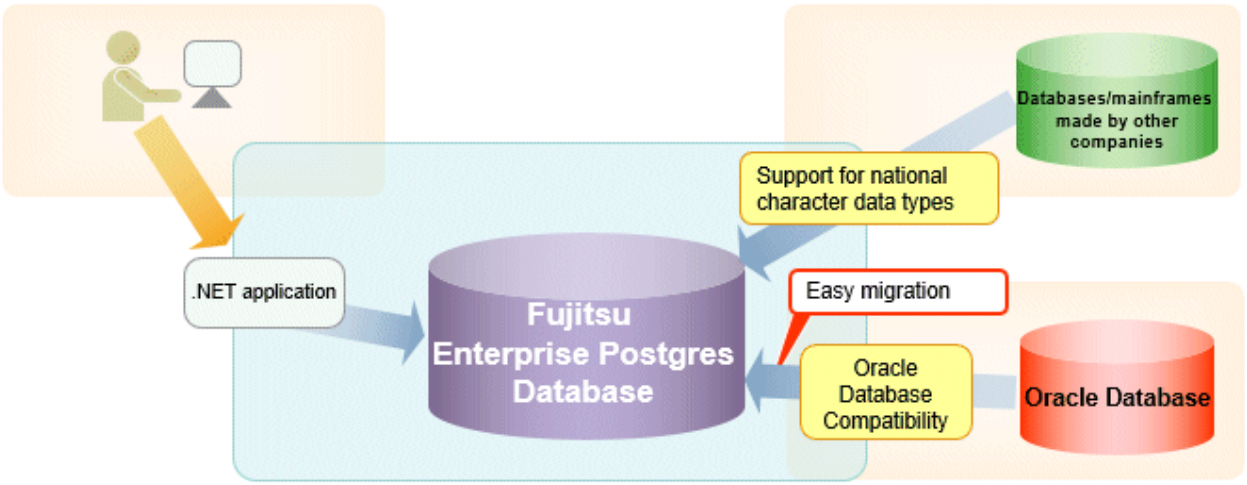
  In order to secure portability from mainframes and databases of other companies, Fujitsu Enterprise Postgres provides data types that support national characters. The national characters are usable from the client application languages.

  Refer to "1.1 Support for National Characters" for details.

- Compatibility with Oracle Databases

  Compatibility with Oracle databases is offered. Use of the compatible features means that the revisions to existing applications can be isolated, and migration to open interfaces is made simpler.

  Refer to "1.2 Compatibility with Oracle Database" for details.



- Application connection switch feature

  The application connection switch feature is provided to enable automatic connection to the target server when there are multiple servers with redundant configurations.

  Refer to "1.3 Application Connection Switch Feature" for details.

- Performance tuning

  The following features are provided to control SQL statement query plans:

  - Optimizer hints
  - Locked statistics

  Refer to "11.1 Enhanced Query Plan Stability" for details.

- Scanning using a Vertical Clustered Index (VCI)

  Scans becomes faster during aggregation of many rows by providing the features below:

  - Vertical clustered index (VCI)
  - In-memory data

  This feature can only be used in Advanced Edition.

  Refer to "Chapter 12 Scan Using a Vertical Clustered Index (VCI)" for details.

1.1 Support for National Characters

NCHAR type is provided as the data type to deal with national characters.

The NCHAR type can be used with Fujitsu Enterprise Postgres pgAdmin.

Point

- NCHAR can only be used when the character set of the database is UTF-8.
- NCHAR can be used in the places where CHAR can be used (function arguments, etc.).
- For applications handling NCHAR type data in the database, the data format is the same as CHAR type. Therefore, applications handling data in NCHAR type columns can also be used to handle data stored in CHAR type columns.

Note

Note the following in order to cast NCHAR type data as CHAR type.

- When comparing NCHAR type data where the length differs, ASCII spaces are used to fill in the length of the shorter NCHAR type data so that it can be processed as CHAR type data.
- Depending on the character set, the data size may increase by between 1.5 and 2 times.
- Use the AS clause to specify "varchar" as the column alias.

1.1.1 Literal

Syntax

```
{ N | n }'[national character [ ...]]'
```

General rules

National character string literals consist of an 'N' or 'n', and the national character is enclosed in single quotation marks ('). Example: N'ABCDEF'

The data type is national character string type.

1.1.2 Data Type

Syntax

```
{ NATIONAL CHARACTER | NATIONAL CHAR | NCHAR } [ VARYING ][(length) ]
```

The data type of the NCHAR type column is as follows:

Data type specification format	Explanation
NATIONAL CHARACTER(*n*) NATIONAL CHAR(*n*) NCHAR(*n*)	National character string with a fixed length of *n* characters This will be the same as (1) if (*n*) is omitted. *n* is a whole number larger than 0.
NATIONAL CHARACTER VARYING(*n*) NATIONAL CHAR VARYING(*n*) NCHAR VARYING(*n*)	National character string with a variable length with a maximum of *n* characters Any length of national character string can be accepted when this is omitted. *n* is a whole number larger than 0.

General rules

NCHAR is the national character string type data type. The length is the number of characters.

The length of the national character string type is as follows:

- When VARYING is not specified, the length of national character strings is fixed and will be the specified length.
- When VARYING is specified, the length of national character strings will be variable.
  In this case, the lower limit will be 0 and the upper limit will be the value specified for length.
- NATIONAL CHARACTER, NATIONAL CHAR, and NCHAR each have the same meaning.

When the national character string to be stored is shorter than the declared upper limit, the NCHAR value is filled with spaces, whereas NCHAR VARYING is stored as is.

The upper limit for character storage is approximately 1GB.

1.1.3 Functions and Operator

Comparison operator

When a NCHAR type or NCHAR VARYING type is specified in a comparison operator, comparison is only possible between NCHAR types or NCHAR VARYING types.

String functions and operators

All of the string functions and operators that can be specified by a CHAR type can also be specified by a NCHAR type. The behavior of these string functions and operators is also the same as with CHAR type.

Pattern matching (LIKE, SIMILAR TO regular expression, POSIX regular expression)

The patterns specified when pattern matching with NCHAR types and NCHAR VARYING types specify the percent sign (%) and the underline (_).

The underline (_) means a match with one national character. The percent sign (%) means a match with any number of national characters 0 or over.

1.2 Compatibility with Oracle Database

The following features have been extended in order to enhance compatibility with Oracle databases:

- Query (external join operator (+), DUAL table)
- Function (DECODE, SUBSTR, NVL)
- Built-in package (DBMS_OUTPUT, UTL_FILE, DBMS_SQL)

Refer to "Chapter 9 Compatibility with Oracle Databases" for information on the features compatible with Oracle databases.

1.3 Application Connection Switch Feature

The application connection switch feature enables automatic connection to the target server when there are multiple servers with redundant configurations.

Refer to " Chapter 10 Application Connection Switch Feature " for information on the application connection switch feature.

1.3.1 Integration with Database Multiplexing

The application connection switch feature is provided to enable automatic connection to the appropriate server when there are multiple servers with redundant configurations.

See

Refer to the Cluster Operation Guide (Database Multiplexing) for information on database multiplexing.

1.4 Notes on Application Compatibility

Fujitsu Enterprise Postgres upgrades contain feature improvements and enhancements that may affect the applications.

Accordingly, note the points below when developing applications, to ensure compatibility after upgrade.

- Checking execution results
- Referencing system catalogs
- Using functions

1.4.1 Checking Execution Results

Refer to SQLSTATE output in messages to check the SQL statements used in applications and the execution results of commands used during development.

See

Refer to Messages for information on the message content and number.

Refer to "PostgreSQL Error Codes" under "Appendixes" in the PostgreSQL Documentation for information on SQLSTATE.

1.4.2 Referencing System Catalogs

System catalogs can be used to obtain information about the Fujitsu Enterprise Postgres system and database objects.

However, system catalogs may change when the Fujitsu Enterprise Postgres version is upgraded. Also, there are many system catalogs that return information that is inherent to Fujitsu Enterprise Postgres.

Accordingly, reference the information schema defined in standard SQL (information_schema) wherever possible. Note also that queries specifying "*" in the selection list must be avoided to prevent columns being added.

See

Refer to "The Information Schema" under "Client Interfaces" in the PostgreSQL Documentation for details.

The system catalog must be referenced to obtain information not found in the information schema. Instead of directly referencing the system catalog in the application, define a view for that purpose. Note, however, that when defining the view, the column name must be clearly specified after the view name.

An example of defining and using a view is shown below.

Example

```
CREATE VIEW my_tablespace_view(spcname) AS SELECT spcname FROM pg_tablespace;
SELECT * FROM my_tablespace_view V1, pg_tables T1 WHERE V1.spcname = T1.tablespace;
```

If changes are made to a system catalog, the user will be able to take action by simply making changes to the view, without the need to make changes to the application.

The following shows an example of taking action by redefining a view as if no changes were made.

The pg_tablespace system catalog is redefined in response to the column name being changed from spcname to spacename.

Example

```
DROP VIEW my_tablespace_view;
CREATE VIEW my_tablespace_view(spcname) AS SELECT spacename FROM pg_tablespace;
```

1.4.3 Using Functions

The default functions provided with Fujitsu Enterprise Postgres enable a variety of operations and manipulations to be performed, and information to be obtained, using SQL statements.

However, it is possible that internal Fujitsu Enterprise Postgres functions, such as those relating to statistical information or for obtaining system-related information, may change as Fujitsu Enterprise Postgres versions are upgraded.

Accordingly, when using these functions, define them as new functions and then use the newly-defined functions in the applications.

An example of defining and using a function is shown below.

Example

```
CREATE FUNCTION my_func(relid regclass) RETURNS bigint LANGUAGE SQL AS 'SELECT
pg_relation_size(relid)';
SELECT my_func(2619);
```

If changes are made to a function, the user will be able to take action by simply redefining the function, without the need to make changes to the application.

The following shows an example of taking action by redefining a function as if no changes were made.

The pg_relation_size function is redefined after arguments are added.

Example

```
DROP FUNCTION my_func(regclass);
CREATE FUNCTION my_func(relid regclass) RETURNS bigint LANGUAGE SQL AS 'SELECT
pg_relation_size(relid,$$main$$)';
```

Chapter 2 JDBC Driver

This section describes how to use JDBC drivers.

2.1 Development Environment

This section describes application development using JDBC drivers and the runtime environment.

2.1.1 Combining with JDK or JRE

Refer to Installation and Setup Guide for Client for information on combining with JDK or JRE where JDBC drivers can operate.

2.2 Setup

This section describes the environment settings required to use JDBC drivers and how to encrypt communication data.

2.2.1 Environment Settings

Configuration of the CLASSPATH environment variable is required as part of the runtime environment for JDBC drivers.

The name of the JDBC driver file is as follows:

```
postgresql-jdbc42.jar
```

Same name whether you use JDK 8, JRE 8, JDK 11, JRE 11, JDK 17 or JRE 17 environment.

The examples below show how to set the CLASSPATH environment variable.

Note that "<*x*>" indicates the product version.

L

- Linux
  - Setting example (TC shell)

```
setenv CLASSPATH /opt/fsepv<x>client64/jdbc/lib/postgresql-jdbc42.jar:${CLASSPATH}
```

  - Setting example (bash)

```
CLASSPATH=/opt/fsepv<x>client64/jdbc/lib/postgresql-jdbc42.jar:$CLASSPATH;export
CLASSPATH
```

W

- Windows (32-bit)
  - Setting example

```
set CLASSPATH=C:\Program Files\Fujitsu\fsepv<x>client32\JDBC\lib\postgresql-
jdbc42.jar;%CLASSPATH%
```

- Windows (64-bit)
  - Setting example (when Fujitsu Enterprise Postgres Client 32-bit is installed)

```
set CLASSPATH=C:\Program Files (x86)\Fujitsu\fsepv<x>client32\JDBC\lib\postgresql-
jdbc42.jar;%CLASSPATH%
```

  - Setting example (when Fujitsu Enterprise Postgres Client 64-bit is installed)

```
set CLASSPATH=C:\Program Files\Fujitsu\fsepv<x>client64\JDBC\lib\postgresql-
jdbc42.jar;%CLASSPATH%
```

2.2.2 Message Language and Encoding System Used by Applications Settings

If the JDBC driver is used, it will automatically set the encoding system on the client to UTF-8, so there is no need to configure this.

See

Refer to "Automatic Character Set Conversion Between Server and Client" in "Server Administration" in the PostgreSQL Documentation for information on encoding systems.

Language settings

You must match the language settings for the application runtime environment with the message locale settings of the database server.

Set language in the "user.language" system property.

Example

Example of running a Java command with system property specified

```
java -Duser.language=en TestClass1
```

2.2.3 Settings for Encrypting Communication Data

When using the communication data encryption feature to connect to the database server, set as follows:

Settings for encrypting communication data for connection to the server

This section describes how to create applications for encrypting communication data.

Set the property of the SSL parameter to "true" to encrypt. The default for the SSL parameter is "false".

If ssl is set to "true", sslmode is internally treated as "verify-full".

Example

- Setting example 1

```
String url = "jdbc:postgresql://sv1/test";
Properties props = new Properties();
props.setProperty("user","fsepuser");
props.setProperty("password","secret");
props.setProperty("ssl","true");
props.setProperty("sslfactory","org.postgresql.ssl.DefaultJavaSSLFactory");
Connection conn = DriverManager.getConnection(url, props);
```

- Setting example 2

```
String url = "jdbc:postgresql://sv1/test?
user=fsepuser&password=secret&ssl=true&sslfactory=org.postgresql.ssl.DefaultJavaSSLF
actory";
Connection conn = DriverManager.getConnection(url);
```

To prevent spoofing of the database server, you need to use the keytool command included with Java to import the CA certificate to the Java keystore. In addition, specify "org.postgresql.ssl.DefaultJavaSSLFactory" for the sslfactory parameter.

Refer to JDK documentation and the Oracle website for details.

Note

There is no need to set the ssl parameter if the connection string of the DriverManager class is specified, or if the sslmode parameter is specified in the data source, such as when the application connection switch feature is used. If the ssl parameter is set, the value in the sslmode parameter will be enabled.

See

Refer to "Secure TCP/IP Connections with SSL" in "Server Administration" in the PostgreSQL Documentation for information on encrypting communication data.

2.3 Connecting to the Database

This section explains how to connect to a database.

- Using the DriverManager Class
- Using the PGConnectionPoolDataSource Class
- Using the PGXADataSource Class

Note

Do not specify "V2" for the "*protocolVersion*" of the connection string.

2.3.1 Using the DriverManager Class

To connect to the database using the DriverManager class, first load the JDBC driver, then specify the connection string as a URI in the API of the DriverManager class.

Load the JDBC driver

Specify org.postgresql.Driver.

Connection string

URI connection is performed as follows:

```
jdbc:postgresql://host:port/database?
user=user&password=password1&loginTimeout=loginTimeout&socketTimeout=socketTimeout
```

Argument	Description
host	Specify the host name for the connection destination. The default is "localhost".
port	Specify the port number for the database server. The default is "27500".
database	Specify the database name.
user	Specify the username that will connect with the database. If this is omitted, the username logged into the operating system that is executing the application will be used.
password	Specify a password when authentication is required.
loginTimeout	Specify the timeout for connections (in units of seconds).

Argument	Description
	Specify a value between 0 and 2147483647. There is no limit set if you set 0 or an invalid value. An error occurs when a connection cannot be established within the specified time.
socketTimeout	Specify the timeout for communication with the server (in units of seconds). Specify a value between 0 and 2147483647. There is no limit set if you set 0 or an invalid value. An error occurs when data is not received from the server within the specified time.

Example

Code examples for applications

```
import java.sql.*;
...
Class.forName("org.postgresql.Driver");
String url = "jdbc:postgresql://sv1:27500/mydb?
user=myuser&password=myuser01&loginTimeout=20&socketTimeout=20";
Connection con = DriverManager.getConnection(url);
```

2.3.2 Using the PGConnectionPoolDataSource Class

To connect to databases using data sources, specify the connection information in the properties of the data source.

Method description

Argument	Description
setServerName	Specify the host name for the connection destination. The default is "localhost".
setPortNumber	Specify the port number for the database server. The default is "27500".
setDatabaseName	Specify the database name.
setUser	Specify the username of the database. By default, the name used will be that of the user on the operating system that is executing the application.
setPassword	Specify a password for server authentication.
setLoginTimeout	Specify the timeout for connections (in units of seconds). Specify a value between 0 and 2147483647. There is no limit set if you set 0 or an invalid value. An error occurs when a connection cannot be established within the specified time.
setSocketTimeout	Specify the timeout for communication with the server (in units of seconds). Specify a value between 0 and 2147483647. There is no limit set if you set 0 or an invalid value. An error occurs when data is not received from the server within the specified time.

Example

Code examples for applications

```
import java.sql.*;
import org.postgresql.ds.PGConnectionPoolDataSource;
...
PGConnectionPoolDataSource source = new PGConnectionPoolDataSource();
source.setServerName("sv1");
source.setPortNumber(27500);
source.setDatabaseName("mydb");
source.setUser("myuser");
source.setPassword("myuser01");
source.setLoginTimeout(20);
source.setSocketTimeout(20);
...
Connection con = source.getConnection();
```

2.3.3 Using the PGXADataSource Class

To connect to databases using data sources, specify the connection information in the properties of the data source.

Method description

Argument	Description
setServerName	Specify the host name for the connection destination. The default is "localhost".
setPortNumber	Specify the port number for the database server. The default is "27500".
setDatabaseName	Specify the database name.
setUser	Specify the username that will connect with the database. If this is omitted, the name used will be that of the user on the operating system that is executing the application.
setPassword	Specify a password when authentication by a password is required.
setLoginTimeout	Specify the timeout for connections. The units are seconds. Specify a value between 0 and 2147483647. There is no limit set if you set 0 or an invalid value. An error occurs when a connection cannot be established within the specified time.
setSocketTimeout	Specify the timeout for communication with the server. The units are seconds. Specify a value between 0 and 2147483647. There is no limit set if you set 0 or an invalid value. An error occurs when data is not received from the server within the specified time.

Example

Code examples for applications

```
import java.sql.*;
import org.postgresql.xa.PGXADataSource;
...
PGXADataSource source = new PGXADataSource();
source.setServerName("sv1");
source.setPortNumber(27500);
source.setDatabaseName("mydb");
source.setUser("myuser");
source.setPassword("myuser01");
source.setLoginTimeout(20);
```

```
source.setSocketTimeout(20);...
Connection con = source.getConnection();
```

2.4 Application Development

This section describes the data types required when developing applications that will be connected with Fujitsu Enterprise Postgres.

2.4.1 Relationship between the Application Data Types and Database Data Types

The following table shows the correspondence between data types in applications and data types in databases.

Data type on the server	Java data type	Data types prescribed by java.sql.Types
character	String	java.sql.Types.CHAR
national character	String	java.sql.Types.NCHAR
character varying	String	java.sql.Types.VARCHAR
national character varying	String	java.sql.Types.NVARCHAR
text	String	java.sql.Types.VARCHAR
bytea	byte[]	java.sql.Types.BINARY
smallint	short	java.sql.Types.SMALLINT
integer	int	java.sql.Types.INTEGER
bigint	long	java.sql.Types.BIGINT
smallserial	short	java.sql.Types.SMALLINT
serial	int	java.sql.Types.INTEGER
bigserial	long	java.sql.Types.BIGINT
real	float	java.sql.Types.REAL
double precision	double	java.sql.Types.DOUBLE
numeric	java.math.BigDecimal	java.sql.Types.NUMERIC
decimal	java.math.BigDecimal	java.sql.Types.DECIMAL
money	String	java.sql.Types.OTHER
date	java.sql.Date	java.sql.Types.DATE
time with time zone	java.sql.Time	java.sql.Types.TIME
time without time zone	java.sql.Time	java.sql.Types.TIME
timestamp without time zone	java.sql.Timestamp	java.sql.Types.TIMESTAMP
timestamp with time zone	java.sql.Timestamp	java.sql.Types.TIMESTAMP
interval	org.postgresql.util.PGInterval	java.sql.Types.OTHER
boolean	boolean	java.sql.Types.BIT
bit	boolean	java.sql.Types.BIT
bit varying	org.postgresql.util.Pgobject	java.sql.Types.OTHER
oid	long	java.sql.Types.BIGINT
xml	java.sql.SQLXML	java.sql.Types.SQLXML

Data type on the server	Java data type	Data types prescribed by java.sql.Types
array	java.sql.Array	java.sql.Types.ARRAY
uuid	java.util.UUID	java.sql.Types.OTHER
point	org.postgresql.geometric.Pgpoint	java.sql.Types.OTHER
box	org.postgresql.geometric.Pgbox	java.sql.Types.OTHER
lseg	org.postgresql.geometric.Pglseg	java.sql.Types.OTHER
path	org.postgresql.geometric.Pgpath	java.sql.Types.OTHER
polygon	org.postgresql.geometric.PGpolygon	java.sql.Types.OTHER
circle	org.postgresql.geometric.PGcircle	java.sql.Types.OTHER
json	org.postgresql.util.PGobject	java.sql.Types.OTHER
Network address type (inet,cidr,macaddr, macaddr8)	org.postgresql.util.PGobject	java.sql.Types.OTHER
Types related to text searches (svector, tsquery)	org.postgresql.util.PGobject	java.sql.Types.OTHER
Enumerated type	org.postgresql.util.PGobject	java.sql.Types.OTHER
Composite type	org.postgresql.util.PGobject	java.sql.Types.OTHER
Range type	org.postgresql.util.PGobject	java.sql.Types.OTHER

Although the getString() method of the ResultSet object can be used for all server data types, it is not guaranteed that it will always return a string in the same format for the same data type.

Strings in a format compatible with the JDBC specifications can be obtained using the Java toString() method of the appropriate data type (for example, getInt(), getTimestamp()) to conform to the data type on the server.

2.4.2 Statement Caching Feature

The statement caching feature caches SQL statements for each individual connection. This means that when an SQL statement with an identical string is next executed, the analysis and creation of the statement can be skipped. This improves performance in cases such as when an SQL statement with an identical string is executed within a loop or method that is executed repeatedly. Furthermore, the statement caching feature can be combined with the connection pooling feature to further enhance performance.

Cache registration controls

You can configure whether to cache SQL statements using the setPoolable(boolean) method of the PreparedStatement class when the statement caching feature is enabled.

Values that can be configured are shown below:

false

SQL statements will not be cached, even when the statement caching feature is enabled.

true

SQL statements will be cached if the statement caching feature is enabled.

2.4.3 Creating Applications while in Database Multiplexing Mode

This section explains points to consider when creating applications while in database multiplexing mode.

- Refer to the Cluster Operation Guide (Database Multiplexing) for information on database multiplexing mode.

- Refer to "Application Development" in the Cluster Operation Guide (PRIMECLUSTER) for points to consider when creating applications using the failover feature integrated with the cluster software.

2.4.3.1 Errors when an Application Connection Switch Occurs and Corresponding Actions

If an application connection switch occurs while in database multiplexing mode, explicitly close the connection and then reestablish the connection or reexecute the application.

The table below shows errors that may occur during a switch, and the corresponding action to take.

State		Error information (*1)	Action
Server failure or Fujitsu Enterprise Postgres system failure	Failure occurs during access	57P01 08006 08007	After the switch is complete, reestablish the connection, or reexecute the application.
	Accessed during system failure	08001	
Switch to the standby server	Switched during access	57P01 08006 08007	
	Accessed during switch	08001	

*1: Return value of the getSQLState() method of SQLException.

Chapter 3 ODBC Driver

This section describes application development using ODBC drivers.

3.1 Development Environment

Applications using ODBC drivers can be developed using ODBC interface compatible applications, such as Access, Excel, and Visual Basic.

Refer to the manuals for the programming languages corresponding to the ODBC interface for information about the environment for development.

Fujitsu Enterprise Postgres supports ODBC 3.5.

3.2 Setup

You need to set up PsqlODBC, which is an ODBC driver, in order to use applications that use ODBC drivers with Fujitsu Enterprise Postgres. PsqlODBC is included in the Fujitsu Enterprise Postgres client package.

The following describes how to register the ODBC drivers and the ODBC data source.

3.2.1 Registering ODBC Drivers

When using the ODBC driver on Linux platforms, register the ODBC driver using the following procedure:

1. Install the ODBC driver manager (unixODBC)

   Information

   - Fujitsu Enterprise Postgres supports unixODBC Version 2.3 or later.

     You can download unixODBC from the following site:

     http://www.unixodbc.org/

   - To execute unixODBC, you must first install libtool 2.2.6 or later.

     You can download libtool from the following website:

     http://www.gnu.org/software/libtool/

   [Note]

   - ODBC driver operation is supported.
   - unixODBC operation is not supported.

2. Register the ODBC drivers

   Edit the ODBC driver manager (unixODBC) odbcinst.ini file.

   Information

   [location of the odbcinst.ini file]

   ```
   unixOdbcInstallDir/etc/odbcinst.ini
   ```

   Set the following content:

Definition name	Description	Setting value
[Driver name]	ODBC driver name	Set the name of the ODBC driver. Select the two strings below that correspond to the application type. Concatenate the strings with no spaces, enclose in "[ ]", and then specify this as the driver name. **Note** The placeholders shown below are enclosed in angle brackets '<>' to avoid confusion with literal text. Do not include the angle brackets in the string. - Application architecture "FujitsuEnterprisePostgres<*fujitsuEnterprisePostgresClientVers*>x64" - Encoding system used by the application - In Unicode (only UTF-8 can be used) "unicode" - Other than Unicode "ansi" Example: The encoding system used by the application is Unicode: "[FujitsuEnterprisePostgres<*fujitsuEnterprisePostgresClientVers*>x64unicode]"
Description	Description of the ODBC driver	Specify a supplementary description for the current data source. Any description may be set.

Driver64	Path of the ODBC driver (64-bit)	Set the path of the ODBC driver (64-bit).
		
		- If the encoding system is Unicode:
		
		`fujitsuEnterprisePostgresClientInstallDir/odbc/lib/psqlodbcw.so`
		
		- If the encoding system is other than Unicode:
		
		`fujitsuEnterprisePostgresClientInstallDir/odbc/lib/psqlodbca.so`

FileUsage	Use of the data source file	Specify 1.
Threading	Level of atomicity secured for connection pooling	Specify 2.

Example

Note that "<x>" indicates the product version.

```
[Fujitsu Enterprise Postgres15x64unicode]
Description = Fujitsu Enterprise Postgres 15 x64 unicode driver
Driver64    = /opt/fsepv<x>client64/odbc/lib/psqlodbcw.so
FileUsage   = 1
Threading   = 2
```

3.2.2 Registering ODBC Data Sources(for Windows(R))

This section describes how to register ODBC data sources.

There are the following two ways to register ODBC data sources on Windows(R).

3.2.2.1 Registering Using GUI

This section describes how to start the [ODBC Data Source Administrator] and register ODBC data sources.

Use the following procedure to register ODBC data sources:

1. Start the [ODBC Data Source Administrator].

   Select [Start] >> [Control Panel] >> [Administrative Tools] >> [ODBC Data Source Administrator].

   Note

   To register data sources for 32-bit applications in Windows(R) for 64-bit, execute the ODBC administrator (odbcad32.exe) for 32-bit, as shown below.

   ```
   %SYSTEMDRIVE%\WINDOWS\SysWOW64\odbcad32.exe
   ```

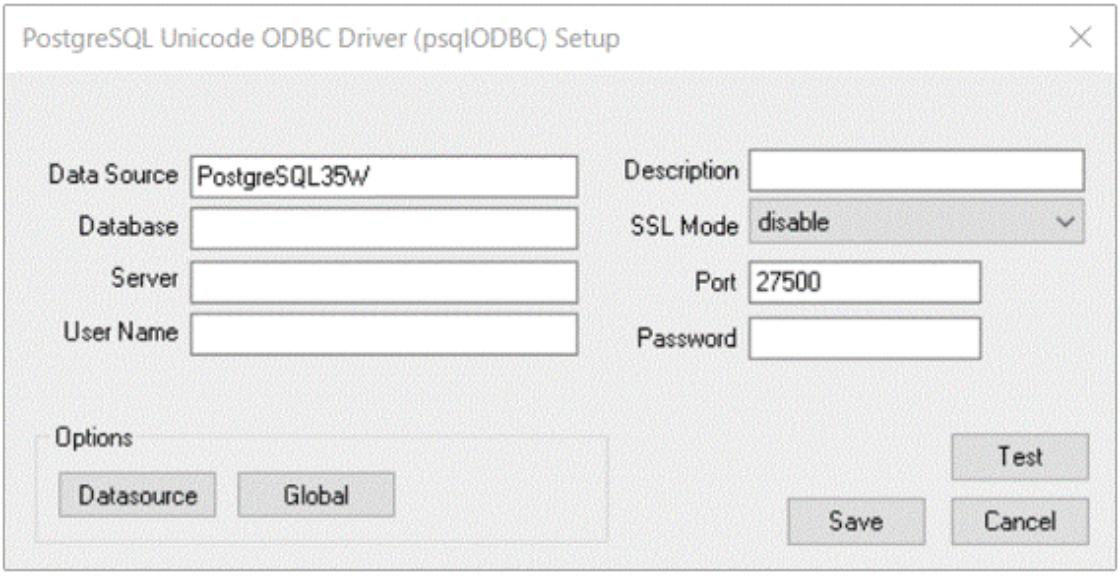
2. When only the current user is to use the ODBC data source, select [User DSN]. When all users using the same computer are to use the ODBC data source, select [System DSN].

3. Click [Add].

4. Select one of the following drivers from the list of available ODBC drivers displayed in [Create New Data Source], and then click [Finish]. The notation "*x*" indicates the version of the Fujitsu Enterprise Postgres client feature.

   - Fujitsu Enterprise Postgres Unicode *x*
     Select this driver if using Unicode as the application encoding system.
   - Fujitsu Enterprise Postgres ANSI *x*
     Select this driver if using other than Unicode as the application encoding system.

5. The [PostgreSQL Unicode ODBC Driver (psqlODBC) Setup] window is displayed. Enter or select the required items, then click [Save].



Set the following content:

Definition name	Setting value
Data Source	Specify the data source name to be registered in the ODBC driver manager. The application will select the name specified here and connect with the Fujitsu Enterprise Postgres database. This parameter cannot be omitted. Specify the following characters up to 32 bytes. - National characters - Alphanumerics - "_", "<", ">", "+", "`", "|", "~", "'" " "&" , """, "#" , "$" , "%" , "-" , "^" , ":" , "/" , "."
Description	Specify a supplementary description for the current data source. Specify characters up to 255 bytes. - National characters - Alphanumerics
Database	Specify the database name to be connected.
SSLMode	Specify to encrypt communications. The default is "disable". The setting values for SSLMode are as follows: - disable: Connect without SSL - allow: Connect without SSL, and if it fails, connect using SSL - prefer: Connect using SSL, and if it fails, connect without SSL - require: Connect always using SSL - verify-ca: Connect using SSL, and use a certificate issued by a trusted CA (*1)

Definition name	Setting value
	- verify-full: Connect using SSL, and use a certificate issued by a trusted CA to verify if the server host name matches the certificate (*1)
Server	Specify the host name of the database server to connect to, using up to 63 bytes. This parameter cannot be omitted.
Port	Specify the port number to be used for remote access. The default value is "27500".
Username(*2)	Specify the user that will access the database.
Password(*2)	Specify the password for the user that will access the database.

*1: If specifying either "verify-ca" or "verify-full", use the system environment variable PGSSLROOTCERT of your operating system to specify the CA certificate file as shown below.

Example:

```
Variable name: PGSSLROOTCERT
Variable value: cACertificateFile
```

*2: In consideration of security, specify the Username and the Password by the application.

3.2.2.2 Registering Using Commands

This section describes how to use commands to register ODBC data sources.

Use the following tools from Microsoft to register ODBC data sources.

- ODBCConf.exe
- Add-OdbcDsn

Refer to the Microsoft Developer Network (MSDN) Library for information on how to use these tools.

When using ODBCConf.exe

ODBCConf.exe is a tool supported on all Windows(R) platforms.

Specification format

```
ODBCConf.exe /A { dataSourceType "odbcDriverName" "optionName=value[|
optionName=value...]"} [/Lv fileName]
```

Refer to the Microsoft MSDN library for information on the format and parameters.

Description

Set the following content:

Definition name	Setting value
Data source type	Specify the data source type. - "CONFIGSYSDSN": A system data source is created. This requires user admin rights. The data source can be used by all users of the same computer. - "CONFIGDSN": A user data source is created. The data source can be used by the current user only.

Definition name	Setting value
	Note When CONFIGSYSDSN is specified as the data source type, it is necessary to execute the command in the command prompt in administrator mode.
ODBC driver name	Specify an ODBC driver name that has already been registered on the system. Specify one of the following. **Note** The placeholders shown below are enclosed in angle brackets '<>' to avoid confusion with literal text. Do not include the angle brackets in the string. - "Fujitsu Enterprise Postgres Unicode *<fujitsuEnterprisePostgresClientVers>*" Specify this driver name if using Unicode as the application encoding system. - "Fujitsu Enterprise Postgres ANSI *<fujitsuEnterprisePostgresClientVers>*" Specify this driver name if using other than Unicode as the application encoding system.
Option name	The following items must be set: - "DSN": Specify the data source name. - "Servername": Specify the host name for the database server. - "Port": Specify the port number for connection to the database - "Database": Specify the database name. Specify the following values as required: - "UID": User ID - "Password": Password - "SSLMode": Specify to encrypt communications. The default is "disable". Refer to the SSLMode explanation in the table under step 5 of "3.2.2.1 Registering Using GUI" for information on how to configure SSLMode.
File Name	You can output process information to a file when creating a data source. This operand can be omitted.

Example

```
ODBCConf.exe /A {CONFIGSYSDSN "Fujitsu Enterprise Postgres Unicode 15" "DSN=odbcconf1|
Servername=sv1|Port=27500|Database=db01|SSLMode=verify-ca"} /Lv log.txt
```

Note

In consideration of security, specify the UID and the Password by the application.

When using Add-OdbcDsn

Add-OdbcDsn is used in the PowerShell command interface.

Specification format

```
Add-OdbcDsn dataSourceName -DriverName "odbcDriverName" -DsnType dataSourceType -
Platform oSArchitecture -SetPropertyValue @("optionName=value" [,"optionName=value"...])
```

Refer to the Microsoft MSDN library for information on the format and parameters.

Description

Set the following content:

Definition name	Setting value
Data source name	Specify any name for the data source name.
ODBC driver name	Specify an ODBC driver name that has already been registered on the system. Specify one of the following. **Note** The placeholders shown below are enclosed in angle brackets '<>' to avoid confusion with literal text. Do not include the angle brackets in the string. - "Fujitsu Enterprise Postgres Unicode *<fujitsuEnterprisePostgresClientVers>*" Specify this driver name if using Unicode as the application encoding system. - "Fujitsu Enterprise Postgres ANSI *<fujitsuEnterprisePostgresClientVers>*" Specify this driver name if using other than Unicode as the application encoding system.
Data source type	Specify the data source type. - "System": A system data source is created. Requires user admin rights. The data source can be used by all users of the same computer. - "User": A user data source is created. The data source can be used by the current user only. **Note** When System is specified as the data source type, it is necessary to execute the command in the administrator mode of the command prompt.
OS architecture	Specify the OS architecture of the system. - "32-bit": 32-bit system - "64-bit": 64-bit system
Option name	The following items must be set: - "Servername": Specify the host name for the database server. - "Port": Specify the port number for connection to the database - "Database": Specify the database name. Specify the following values as required: - "SSLMode": Specify to encrypt communications. The default is "disable". Refer to the SSLMode explanation in the table under step 5 of "3.2.2.1 Registering Using GUI" for information on how to configure SSLMode.

Definition name	Setting value
	Note When using Add-OdbcDsn, the strings "UID" and "Password" cannot be set as option names. These can only be used when using ODBCConf.exe.

Example

```
Add-OdbcDsn odbcps1 -DriverName "Fujitsu Enterprise Postgres Unicode 15" -DsnType System -
Platform 32-bit -SetPropertyValue @("Servername=sv1", "Port=27500", "Database=db01",
"SSLMode=verify-ca")
```

3.2.3 Registering ODBC Data Sources(for Linux)

This section describes how to register ODBC data sources on Linux.

1. Register the data sources

   Edit the odbc.ini definition file for the data source.

   Information

   Edit the file in the installation directory for the ODBC driver manager (unixODBC)

   ```
   unixOdbcInstallDir/etc/odbc.ini
   ```

   Or

   Create a new file in the HOME directory

   ```
   ~/.odbc.ini
   ```

   Point

   If *unixOdbcInstallDir* is edited, these will be used as the shared settings for all users that log into the system. If created in the HOME directory (~/), the settings are used only by the single user.

   Set the following content:

Definition name	Setting value
[Data source name]	Set the name for the ODBC data source.
Description	Set a description for the ODBC data source. Any description may be set.
Driver	Set the following as the name of the ODBC driver. Do not change this value. Select the two strings below that correspond to the application type. Concatenate the strings with no spaces and then specify this as the driver name.

Definition name	Setting value
	Note The placeholders shown below are enclosed in angle brackets '<>' to avoid confusion with literal text. Do not include the angle brackets in the string. - Application architecture "Fujitsu Enterprise Postgres *<fujitsuEnterprisePostgresClientVers>*x64" - Encoding system used by the application - In Unicode (only UTF-8 can be used) "unicode" - Other than Unicode "ansi" Example: The encoding system used by the application is Unicode: "Fujitsu Enterprise Postgres *<fujitsuEnterprisePostgresClientVers>*x64unicode"
Database	Specify the database name to be connected.
Servername	Specify the host name for the database server.
Username	Specify the user ID that will connect with the database.
Password	Specify the password for the user that will connect to the database.
Port	Specify the port number for the database server. The default is "27500".
SSLMode	Specify the communication encryption method. The setting values for SSLMode are as follows: - disable: Connect without SSL - allow: Connect without SSL, and if it fails, connect using SSL - prefer: Connect using SSL, and if it fails, connect without SSL - require: Connect always using SSL - verify-ca: Connect using SSL, and use a certificate issued by a trusted CA (*1) - verify-full: Connect using SSL, and use a certificate issued by a trusted CA to verify if the server host name matches the certificate (*1)
ReadOnly	Specify whether to set the database as read-only. - 1: Set read-only - 0: Do not set read-only

*1: If specifying either "verify-ca" or "verify-full", use the environment variable PGSSLROOTCERT to specify the CA certificate file as shown below.

Example

```
export PGSSLROOTCERT=cACertificateFileStorageDir/root.crt
```

Example

```
[MyDataSource]
Description    = Fujitsu Enterprise Postgres
```

```
Driver          = Fujitsu Enterprise Postgres15x64unicode
Database        = db01
Servername      = sv1
Port            = 27500
ReadOnly        = 0
```

Note

In consideration of security, specify the UserName and the Password by the application.

2. Configure the environment variable settings

   To execute applications that use ODBC drivers, all of the following settings must be configured in the LD_LIBRARY_PATH environment variable:

   - *unixOdbcInstallDir(*1)*/lib
   - *libtoolInstallDir(*1)*/lib

   *1: If the installation directory is not specified when unixODBC and libtool are installed, they will be installed in /usr/local.

3.2.4 Message Language and Encoding System Used by Applications Settings

This section explains the language settings for the application runtime environment and the encoding settings for the application.

Language settings

You must match the language settings for the application runtime environment with the message locale settings of the database server.

Messages output by an application may include text from messages sent from the database server. In the resulting text, the text of the application message will use the message locale of the application, and the text of the message sent by the database server will use the message locale of the database server. If the message locales do not match, more than one language or encoding system will be used. Moreover, if the encoding systems do not match, characters in the resulting text can be garbled.

L

- Linux

  Set the locale for messages (LC_MESSAGES category) to match the message locale of the database server. This can be done in a few different ways, such as using environment variables. Refer to the relevant manual of the operating system for information on the setlocale function.

  Example

  Example of specifying "en_US.UTF-8" with the setlocale function

  ```
  setlocale(LC_ALL,"en_US.UTF-8");
  ```

  Specifying the locale of the LC_ALL category propagates the setting to LC_MESSAGE.

W

- Windows(R)

  Align the locale of the operating system with the message locale of the database server.

Encoding System Settings

Ensure that the encoding system that is embedded in the application and passed to the database, and the encoding system setting of the runtime environment, are the same. The encoding system cannot be converted correctly on the database server.

Use one of the following methods to set the encoding system for the application:

- Set the PGCLIENTENCODING environment variable in the runtime environment.
- Set the client_encoding keyword in the connection string.
- Use the PQsetClientEncoding function.

See

Refer to "Supported Character Sets" in "Server Administration" in the PostgreSQL Documentation for information on the strings that represent the encoding system that can be set.

For example, when using "Unicode" and "8 bit", set the string "UTF8".

Example

Setting the "PGCLIENTENCODING" environment variable

L

An example of setting when the encoding of the client is "UTF8" (Bash)

```
> PGCLIENTENCODING=UTF8; export PGCLIENTENCODING
```

W

An example of setting when the encoding of the client is "UTF8"

```
> set PGCLIENTENCODING=UTF8
```

Note

Text may be garbled when outputting results to the command prompt. Review the font settings for the command prompt if this occurs.

3.3 Connecting to the Database

Refer to the manual for the programming language corresponding to the ODBC interface, i.e. Access, Excel, or Visual Basic, for example.

3.4 Application Development

This section describes how to develop applications using ODBC drivers.

W

3.4.1 Compiling Applications (for Windows (R))

Refer to the manual for the programming language corresponding to the ODBC interface, i.e. Access, Excel, or Visual Basic, for example.

Note

The cl command expects input to be a program that uses one of the following code pages, so convert the program to these code pages and then compile and link it (refer to the Microsoft documentation for details).

- ANSI console code pages (example: UTF8)
- UTF-16 little-endian with or without BOM (Byte Order Mark)
- UTF-16 big-endian with or without BOM
- UTF-8 with BOM

The cl command converts strings in a program to an ANSI console code page before generating a module, so the data sent to and received from the database server becomes an ANSI console code page. Therefore, set the coding system corresponding to the ANSI console code page as the coding system of the client.

Refer to "Character Set Support" in "Server Administration" in the PostgreSQL Documentation for information on how to set the client encoding system.

L

3.4.2 Compiling Applications (for Linux)

Specify the following options when compiling applications.

Table 3.1 Include file and library path

Option	How to specify the option
Path of the include file	-I *unixOdbc64bitIncludeFileDir*
Path of the library	-L *unixOdbc64bitLibraryDir*

Table 3.2 ODBC library

Type of library	Library name
Dynamic library	libodbc.so

Note

Specify -m64 when creating a 64-bit application.

Example

The following are examples of compiling ODBC applications:

```
gcc -m64 -I/usr/local/include(*1) -L/usr/local/lib(*1) -lodbc testproc.c -o testproc
```

*1: This is an example of building and installing from the source without specifying an installation directory for unixODBC. If you wish to specify a location, set the installation directory.

3.4.3 Creating Applications While in Database Multiplexing Mode

This section explains points to consider when creating applications while in database multiplexing mode.

See

- Refer to the Cluster Operation Guide (Database Multiplexing) for information on database multiplexing mode.

- Refer to "Application Development" in the Cluster Operation Guide (PRIMECLUSTER) for points to consider when creating applications using the failover feature integrated with the cluster software.

3.4.3.1 Errors when an Application Connection Switch Occurs and Corresponding Actions

If an application connection switch occurs while in database multiplexing mode, explicitly close the connection and then reestablish the connection or reexecute the application.

The table below shows errors that may occur during a switch, and the corresponding action to take.

State		Error information (*1)	Action
Server failure or Fujitsu Enterprise Postgres system failure	Failure occurs during access	57P01 08S01	After the switch is complete, reestablish the connection, or reexecute the application.
	Accessed during system failure	08001	
Switch to the standby server	Switched during access	57P01 08S01	
	Accessed during switch	08001	

*1: Return value of SQLSTATE.

Chapter 4 .NET Data Provider

This chapter describes how to configure for the purpose of creating .NET applications with Visual Studio.

4.1 Development Environment

.NET Data Provider can operate in the following environments:

Environment required for development and operation of .NET applications	.NET 6.0 .NET 5.0 .NET Core 3.1 .NET Standard 2.0 .NET Standard 2.1
Integrated development environment for applications running in a .NET environment	Visual Studio 2022 Visual Studio 2019 Visual Studio 2017 Visual Studio 2015
Available development languages	C# Visual Basic .NET

4.2 Setup

This section explains how to set up .NET Data Provider and Entity Framework Core.

4.2.1 Setting Up .NET Data Provider

The .NET Data Provider of Fujitsu Enterprise Postgres can be installed as a local NuGet package. Follow the steps below to install.

Location of NuGet package

The NuGet package are stored in the following location. "<*x*>" indicates the product version.

```
fujitsuEnterprisePostgresClientInstallDir\DOTNET\Npgsql.<x>.0.0.nupkg
```

Add a local package source

In Visual Studio, add a NuGet local package source if one does not exist.

1. Click [Tools] >> [Options] >> [NuGet Package Manager], and then select [Package Sources].
2. Click [+] in the upper-right corner, and then set [Name] to "Local Package Source".
3. Click [⋯] and navigate to the folder above. Select this folder, and then click [OK].

Install the NuGet package

In Visual Studio, install the NuGet package from the local package source.

1. Click [Tools] >> [NuGet Package Manager] >> [Manage NuGet Packages for Solution].
2. In the upper-right corner, select "Local Package Source" from [Package Source].
3. Once the local package source is set, all available NuGet packages in this local location will be displayed. Select "Npgsql", and then select the projects for which this package is to be installed.
4. Click [Install].

4.2.2 Setting Up .NET Data Provider Type Plugins

Fujitsu Enterprise Postgres .NET Data Provider now comes packaged with 4 Type Plugins that provide additional support for more data type mappings (eg date time support via the Npgsql.NodaTime). The plugins modify how Npgsql maps the PostgreSQL values to CLR types.

The type plugins are available for installation as a local NuGet packages. There are 4 pacakages available for installation. "<*x*>" indicates the product version.

- Npgsql.NodaTime.<*x*>.0.0.nupkg
- Npgsql.Json.NET.<*x*>.0.0.nupkg
- Npgsql.NetTopologySuite.<*x*>.0.0.nupkg: Spatial type(*1)
- Npgsql.GeoJSON.<*x*>.0.0.nupkg: Spatial type(*1)

*1: Please note that the Spatial Type plugins require the PostGIS extension installed on the server.

Also, Refre to "Type Plugins" and "Additional Notes on each Type Plugin" on the each type plugins.

To install any of the plugins please follow the procedure below:

Location of NuGet package

The Plugin NuGet package are stored in the following location:

```
fujitsuEnterprisePostgresClientInstallDir\DOTNET\Npgsql.*.<x>.0.0.nupkg
```

Add a local package source

In Visual Studio, add a NuGet local package source if one does not exist.

1. Click [Tools] >> [Options] >> [NuGet Package Manager], and then select [Package Sources].
2. Click [+] in the upper-right corner, and then set [Name] to "Local Package Source".
3. Click [···] and navigate to the folder above. Select this folder, and then click [OK].

Install the NuGet package

In Visual Studio, install the NuGet package from the local package source.

1. Click [Tools] >> [NuGet Package Manager] >> [Manage NuGet Packages for Solution].
2. In the upper-right corner, select "Local Package Source" from [Package Source].
3. Once the local package source is set, all available NuGet packages in this local location will be displayed. Select the plugin to be installed (eg "Npgsql.NodaTime", and then select the projects for which this package is to be installed.
4. Click [Install].

4.2.3 Setting Up Entity Framework Core

Entity Framework Core is supplied as a NuGet package file. To install it locally, follow the procedure below.

Location of NuGet package

The EntityFrameworkCore.Npgsql NuGet package is stored in the following location:

```
fujitsuEnterprisePostgresClientInstallDir\DOTNET\Npgsql.EntityFrameworkCore.PostgreSQL.
6.0.7.nupkg
```

Add a local package source

In Visual Studio, add a NuGet local package source if one does not exist.

1. Click [Tools] >> [Options] >> [NuGet Package Manager], and then select [Package Sources].

2. Click [+] in the upper-right corner, and then set [Name] to "Local Package Source".
3. Click [⋯] and navigate to the folder above. Select this folder, and then click [OK].

Install the NuGet package

In Visual Studio, install the NuGet package from the local package source.

1. Click [Tools] >> [NuGet Package Manager] >> [Manage NuGet Packages for Solution].
2. In the upper-right corner, select "Local Package Source" from [Package Source].
3. Once the local package source is set, all available NuGet packages in this local location will be displayed. Select "Npgsql.EntityFrameworkCore.PostgreSQL", and then select the projects for which this package is to be installed.
4. Click [Install].

4.2.4 Setting Up Npgsql.OpenTelemetry

Fujitsu Enterprise Postgres .NET Data Provider comes with a plug-in that supports tracing by the observability framework OpenTelemetry. This will allow Npgsql to emit trace data.

OpenTelemetry plugin is supplied as a local NuGet package. To install it locally, follow the procedure below.

Location of NuGet package

The NuGet packages for type plugins are stored in the following location. "<*x*>" indicates the product version.

```
fujitsuEnterprisePostgresClientInstallDir\DOTNET\Npgsql.OpenTelemetry.<x>.0.0.nupkg
```

Add a local package source

In Visual Studio, add a NuGet local package source if one does not exist.

1. Click [Tools] >> [Options] >> [NuGet Package Manager], and then select [Package Sources].
2. Click [+] in the upper-right corner, and then set [Name] to "Local Package Source".
3. Click [⋯] and navigate to the folder above. Select this folder, and then click [OK].

Install the NuGet package

In Visual Studio, install the NuGet package from the local package source.

1. Click [Tools] >> [NuGet Package Manager] >> [Manage NuGet Packages for Solution].
2. In the upper-right corner, select "Local Package Source" from [Package Source].
3. Once the local package source is set, all available NuGet packages in this local location will be displayed. Select "Npgsql.OpenTelemetry", and then select the projects for which this package is to be installed.
4. Click [Install].

4.2.5 Message Language Settings

You must match the language settings for the application runtime environment with the message locale settings of the database server.

Set language using the "System.Globalization.CultureInfo.CreateSpecificCulture" method.

Example

Code example for changing the locale in a C# application

```
System.Threading.Thread.CurrentThread.CurrentUICulture =
    System.Globalization.CultureInfo.CreateSpecificCulture("en");
```

4.3 Connecting to the Database

This section explains how to connect to a database.

- Using NpgsqlConnection
- Using NpgsqlConnectionStringBuilder

4.3.1 Using NpgsqlConnection

Connect to the database by specifying the connection string.

Example

Code examples for applications

```
using Npgsql;

NpgsqlConnection conn = new NpgsqlConnection("Server=sv1;Port=27500;Database=mydb;
Username=myuser;Password=myuser01; Timeout=20;CommandTimeout=20;");
```

Refer to "4.3.3 Connection String" for information on database connection strings.

4.3.2 Using NpgsqlConnectionStringBuilder

Generate connection strings by specifying the connection information in the properties of the NpgsqlConnectionStringBuilder object.

Example

Code examples for applications

```
using Npgsql;

NpgsqlConnectionStringBuilder sb = new NpgsqlConnectionStringBuilder();
sb.Host = "sv1";
sb.Port = 27500;
sb.Database = "mydb";
sb.Username = "myuser";
sb.Password = "myuser01";
sb.Timeout = 20;
sb.CommandTimeout = 20;
NpgsqlConnection conn = new NpgsqlConnection(sb.ConnectionString);
```

Refer to "4.3.3 Connection String" for information on database connection strings.

4.3.3 Connection String

Specify the following connection information to connect to the database.

```
Server=127.0.0.1;Port=27500;Database=mydb;Username=myuser;Password=myuser01;...;
       (1)           (2)           (3)          (4)             (5)      (6)
```

(1) Specify the host name or IP address of the server to be connected. This must be specified.

(2) Specify the port number for the database server. The default is "27500".

(3) Specify the database name to be connected.

(4) Specify the username that will connect with the database.

(5) Specify the password for the user that will connect to the database.

(6) Refer to the following for information on how to specify other connection information.

The table below shows keywords that are available to specify in the connection string in .NET Data Provider (Npgsql):

Note that some settings require care if using an Oracle database-compatible feature (refer to "9.2.3 Notes when Integrating with the Interface for Application Development" for details).

Basic connection

Keyword	Description	Default
Host	Specify the host name of the server to be connected. Multiple hosts may be specified. A host name must be specified.	
Port	Specify the TCP port number of the server.	27500
Database	Specify the database name to connect to.	Same as Username
Username	Specify the username for the username that will connect to the database. Not required if using Integrated Security.	PGUSER
Password	Specify the password for the username that will connect to the database. Not required if using Integrated Security.	PGPASSWORD
Passfile	Specify the path of the password file (PGPASSFILE) to obtain the password.	PGPASSFILE

Security and encryption

Keyword	Description	Default
SSL Mode	Specify one of the following values to indicate whether or not to use SSL, depending on your server's support. Disable: No SSL connection is made. Allow: SSL connection will be made only if the server requires it. Prefer: Make an SSL connection if possible. Require: Perform SSL connection. It does not verify the authenticity of Connection Server at this time. When specifying this, you must specify "Trust Server Certificate=true". VerifyCA: Make an SSL connection. At this time, verify the authenticity of the connection server. VerifyFull: Make an SSL connection. This time it verifies the authenticity of the Connection Server and verifies that it is the server you specified.	Prefer
Trust Server Certificate	Specify whether to trust the server without verifying the server certificate.	false
SSL Certificate	Specify the location of the client certificate sent to the server.	PGSSLCERT
SSL Key	Specify the location of the client key for the client certificate sent to the server.	PGSSLKEY
SSL Password	Specify the password for the client certificate key.	
Root Certificate	Specify the location of the CA certificate used to validate server certificates.	PGSSLROOTCERT

Keyword	Description	Default
Check Certificate Revocation	Specify whether to check the certificate revocation list during authentication.	false
Integrated Security	Specify whether to log in using integrated security (GSS/SSPI).	false
Persist Security Info	Gets or sets a Boolean value that indicates whether security-sensitive information, such as passwords, should not be returned as part of a connection if it is connected or has ever been connected.	false
Kerberos Service Name	Specify the Kerberos service name used for authentication.	postgres
Include Realm	Specify the Kerberos realm used for authentication.	
Include Error Detail	When enabled, database error and notification details are included in PostgresException.Detail and PostgresNotice.Detail. These can contain confidential information.	false
Log Parameters	When enabled, parameter values are logged when the command is executed.	false

Pooling

Keyword	Description	Default
Pooling	Specify whether to use connection pooling.	true
Minimum Pool Size	Minimum connection pool size.	0
Maximum Pool Size	Maximum size of the connection pool.	100
Connection Idle Lifetime	Specify the time (in seconds) to wait before disconnecting idle connections in the pool when the number of all connections exceeds the Minimum Pool Size.	300
Connection Pruning Interval	Specify the number of seconds the pool waits before attempting to prune an idle connection that has exceeded its lifetime. See Connection Idle Lifetime.	10
ConnectionLifetime	Specify the total maximum lifetime of the connection in seconds. Connections beyond this value are discarded rather than returned from the pool. This is useful in cluster configurations that force load balancing between running servers and servers that have just come online.	0 (disabled)

Timeouts and keepalive

Keyword	Description	Default
Timeout	Specify the amount of time (in seconds) to wait before terminating the attempt and generating an error during connection establishment.	15
Command Timeout	Specify how long to wait (in seconds) while executing a command before terminating the attempt and generating an error. Set to 0 for infinity.	30
Internal Command Timeout	Specify how long to wait (in seconds) before terminating the attempt and generating an error when attempting to execute an internal command. -1 uses CommandTimeout, 0 means no timeout.	-1

Keyword	Description	Default
Cancellation Timeout	Specify the number of milliseconds to wait while trying to read a response to a cancel request for a timed-out or canceled query before terminating the attempt and generating an error. -1 skips the wait, 0 means infinite wait.	2000
Keepalive	Npgsql will send his keep-alive queries if the connection has been inactive for more than the number of seconds specified here.	0 (disabled)
Tcp Keepalive	Specify whether TCP keepalives are used by the system default if no override is specified.	false
Tcp Keepalive Time	A TCP keep-alive query is sent when the connection inactivity exceeds the number of milliseconds specified here. Use of this option is not recommended. We recommend using Keepalive. This is supported on Windows only.	0 (disabled)
Tcp Keepalive Interval	Specify the interval (in milliseconds) between successive keepalive packets sent if no acknowledgment is received. Tcp KeepAlive Time must also be non-zero. This is supported on Windows only.	TCP Keepalive Time value

Performance

Keyword	Description	Default
Max Auto Prepare	Maximum number of SQL statements that can be automatically prepared at any given time. When this number is exceeded, the last used statement is reused. A value of 0 disables autoprepare.	0
Auto Prepare Min Usages	A SQL statement is automatically prepared when its usage count exceeds the value specified here.	5
Read Buffer Size	The size of the internal buffer Npgsql uses when reading. Increasing this value may improve performance when transferring large amounts of data from the database.	8192
Write Buffer Size	Determines the size of the internal buffer Npgsql uses when writing. Increasing this value may improve performance when transferring large amounts of data to the database.	8192
Socket Receive Buffer Size	The size of the socket receive buffer.	System dependent
Socket Send Buffer Size	The size of the socket send buffer.	System dependent
No Reset On Close	When true, do not reset the connection state when the connection is returned to the pool. In some cases, this improves performance at the cost of leak conditions. Use it only if your benchmark shows performance improvements.	false

Failover and load balancing

Keyword	Description	Default
Target Session Attributes	Specifies the server type to preferentially connect to. any: Allows any successful connection. primary: Connect to the primary server. standby: Connect to the standby server.	PGTARGETSESSIONATTRS, Any

Keyword	Description	Default
	prefer-primary: Attempt to connect to the primary server first, but retry in any mode if none of the listed hosts has a primary server. prefer-standby: Attempt to connect to the standby server first, but retry in any mode if there is no standby server on the listed host. read-write: Connect to a server whose default transaction mode is read/write. read-only: Connect to a server whose default transaction mode is read-only.	
Load Balance Hosts	Specify whether to perform load balancing among multiple hosts using the round-robin method.	false
Host Recheck Seconds	Controls how long a host's cached state is considered valid.	10
Enable Fdw Acs	Specifies whether to switch the connection destination type of the data node with on or off when using the connection routing function. This cannot be set to on when "Target Session Attributes=prefer-primary" is specified.	off

Others

Keyword	Description	Default
Options	Specify valid database connection options. (Example: Options=-c synchronous_commit=local)	PGOPTIONS
Application Name	Specify the application name sent to the backend when opening a connection.	
Enlist	Specify whether the connection should participate in transactions declared in the transaction scope.	true
Search Path	Sets the schema search path.	None
Client Encoding	Gets or sets the client_encoding parameter.	PGCLIENTENCODING
Encoding	Gets or sets the .NET encoding used to encode/decode string data in the database.	UTF8
Timezone	Gets or sets the session timezone.	PGTZ
EF Template Database	A database template that is specified when creating a database with Entity Framework.	template1
EF Admin Database	The managed database that you specify when creating and deleting databases in Entity Framework.	template1
Load Table Composites	Specify whether to load complex type definitions for tables in addition to independent complex types.	false
Array Nullability Mode	Sets how to return an array of value types when requested as an object instance. Possible values are: Never: Always return as a non-nullable array. Always: Always return as a nullable array. PerInstance: The returned array is determined at runtime.	Never

4.4 Application Development

4.4.1 Data Types

A variety of data types can be used with Fujitsu Enterprise Postgres.

List of supported data types

Data Type
boolean
smallint
integer
bigint
real
double precision
numeric
money
text
character varying
character
national character varying
national character
citext
json
jsonb
xml
uuid
bytea
timestamp without time zone
timestamp with time zone
date
time without time zone(*1)
time with time zone(*1)
interval(*2)
cidr
inet
macaddr
tsquery
tsvector
bit(1)
bit(n)
bit varying

Data Type
point
lseg
path
polygon
line
circle
box
hstore
oid
xid
cid
oidvector
name
(internal) char
geometry (PostGIS)
record
composite types
range types
multirange types (PG14)
enum types
array types

*1: As shown below, "time with time zone" and "time without time zone" values display the date portion as additional information. However, the actual data comprises the time data only, so with the exception of this displayed format, there are no other resulting issues.

Example:

Composition of table (t1)

col1 (time with time zone)	col2 (time without time zone)
1/01/0001 10:21:30 +08:00	10:21:30
1/01/0001 23:34:03 +08:00	23:34:03
1/01/0001 17:23:54 +08:00	17:23:54

"time with time zone" values display a fixed value of “2/01/0001” in the date portion, while "time without time zone" values display just the time data.

```
SELECT *
  FROM t1;
 col1                       |    col2
----------------------------+--------------------
 1/01/0001 10:21:30 +08:00 | 10:21:30
 1/01/0001 23:34:03 +08:00 | 23:34:03
 1/01/0001 17:23:54 +08:00 | 17:23:54
```

*2: The format is d.hh:mm:ss, where d is an integer and hh:mm:ss is a maximum of 23.59.59 (23 hours, 59 minutes, and 59 seconds).

4.4.2 Relationship between Application Data Types and Database Data Types

The data types available for SQL data types are as follows:

Read mappings

PostgreSQL Type	Default.NET Type	Non-default Type
boolean	bool	
smallint	short	byte, sbyte, int, long, float, double, decimal
integer	int	byte, short, long, float, double, decimal
bigint	long	long, byte, short, int, float, double, decimal
real	float	double
double precision	double	
numeric	decimal	byte, short, int, long, float, double, BigInteger
money	decimal	
text	string	char[]
character varying	string	char[]
character	string	char[]
national character varying	string	char[]
national character	string	char[]
citext	string	char[]
json	string	char[]
jsonb	string	char[]
xml	string	char[]
uuid	Guid	
bytea	byte[]	
timestamp without time zone	DateTime (Unspecified)	
timestamp with time zone	DateTime (Utc)(*1)	DateTimeOffset (Offset=0)(*2)
date	DateTime	DateOnly
time without time zone	TimeSpan	TimeOnly
time with time zone	DateTimeOffset	
interval	TimeSpan(*3)	NpgsqlInterval
cidr	(IPAddress, int)	NpgsqlInet
inet	IPAddress	NpgsqlInet, (IPAddress, int)
macaddr	PhysicalAddress	
tsquery	NpgsqlTsQuery	
tsvector	NpgsqlTsVector	
bit(1)	bool	BitArray

PostgreSQL Type	Default.NET Type	Non-default Type
bit(n)	BitArray	
bit varying	BitArray	
point	NpgsqlPoint	
lseg	NpgsqlLSeg	
path	NpgsqlPath	
polygon	NpgsqlPolygon	byte, sbyte, int, long, float, double, decimal
line	NpgsqlLine	
circle	NpgsqlCircle	
box	NpgsqlBox	
hstore	Dictionary<string, string>	
oid	uint	
xid	uint	
cid	uint	
oidvector	uint[]	
name	string	char[]
(internal) char	char	byte, short, int, long
geometry (PostGIS)	PostgisGeometry	
record	object[]	
composite types	T	
range types	NpgsqlRange<TElement>	
multirange types (PG14)	NpgsqlRange<TElement>[]	
enum types	TEnum	
array types	Array (of element type)	

*1: When Npgsql.EnableLegacyTimestampBehavior is enabled, reading a timestamp with time zone returns a Local DateTime instead of Utc.

*2: When Npgsql.EnableLegacyTimestampBehavior is enabled, reading a timestamp with time zone as a DateTimeOffset returns a local offset based on the timezone of the server where Npgsql is running.

*3: PostgreSQL intervals with month or year components cannot be read as TimeSpan. Consider using NodaTime's Period type, or NpgsqlInterval.

Write mappings

PostgreSQL Type	Default.NET Type	Non-default.NET Type	NpgsqlDbType	DbType
boolean	bool		Boolean	Boolean
smallint	short, byte, sbyte		Smallint	Int16
integer	int		Integer	Int32
bigint	long		Bigint	Int64
real	float		Real	Single
double precision	double		Double	Double

PostgreSQL Type	Default.NET Type	Non-default.NET Type	NpgsqlDbType	DbType
numeric	decimal, BigInteger		Numeric	Decimal, VarNumeric
money		decimal	Money	Currency
text	string, char[], char		Text	String, StringFixedLength, AnsiString, AnsiStringFixedLength
character varying		string, char[], char	Varchar	
character		string, char[], char	Char	
citext		string, char[], char	Citext	
json		string, char[], char	Json	
jsonb		string, char[], char	Jsonb	
xml		string, char[], char	Xml	
uuid	Guid		Uuid	
bytea	byte[]	ArraySegment<byte>	Bytea	Binary
timestamp without time zone	DateTime (Utc) (*1), DateTimeOffset		TimestampTz	DateTime(*2), DateTimeOffset
timestamp with time zone	DateTime (Local/ Unspecified) (*1)		Timestamp	DateTime2
date	DateOnly	DateTime	Date	Date
time without time zone	TimeOnly	TimeSpan	Time	Time
time with time zone		DateTimeOffset	TimeTz	
interval	TimeSpan	NpgsqlInterval	Interval	
cidr		ValueTuple<IPAddress, int>, IPAddress	Cidr	
inet	IPAddress	ValueTuple<IPAddress, int>	Inet	
macaddr	PhysicalAddress		MacAddr	
tsquery	NpgsqlTsQuery		TsQuery	
tsvector	NpgsqlTsVector		TsVector	
bit		bool, BitArray, string	Bit	
bit varying	BitArray	bool, BitArray, string	Varbit	
point	NpgsqlPoint		Point	
lseg	NpgsqlLSeg		LSeg	
path	NpgsqlPath		Path	

PostgreSQL Type	Default.NET Type	Non-default.NET Type	NpgsqlDbType	DbType
polygon	NpgsqlPolygon		Polygon	
line	NpgsqlLine		Line	
circle	NpgsqlCircle		Circle	
box	NpgsqlBox		Box	
hstore	IDictionary<string, string>		Hstore	
oid		uint	Oid	
xid		uint	Xid	
cid		uint	Cid	
oidvector		uint[]	Oidvector	
name		string, char[], char	Name	
(internal) char		byte	InternalChar	
composite types	Pre-mapped type		Composite	
range types	NpgsqlRange<TSubtype>		Range | NpgsqlDbType	
enum types	Pre-mapped type		Enum	
array types	T[], List<T>		Array | NpgsqlDbType	

*1: UTC DateTime is written as timestamp with time zone, Local/Unspecified DateTimes are written as timestamp without time zone. When Npgsql.EnableLegacyTimestampBehavior is enabled, DateTime is always written as timestamp without time zone.

*2: When Npgsql.EnableLegacyTimestampBehavior is enabled, DbType.DateTime is mapped to timestamp without time zone.

4.4.3 Creating Applications while in Database Multiplexing Mode

This section explains points to consider when creating applications while in database multiplexing mode.

- Refer to the Cluster Operation Guide (Database Multiplexing) for information on database multiplexing mode.
- Refer to "Application Development" in the Cluster Operation Guide (PRIMECLUSTER) for points to consider when creating applications using the failover feature integrated with the cluster software.

4.4.3.1 Errors when an Application Connection Switch Occurs and Corresponding Actions

If an application connection switch occurs while in database multiplexing mode, explicitly close the connection and then reestablish the connection or reexecute the application.

The table below shows errors that may occur during a switch, and the corresponding action to take.

State		Error information	Action
Server failure or Fujitsu Enterprise Postgres system failure	Failure occurs during access	57P01 (*1) Empty string (*1)	After the switch is complete, reestablish the connection, or reexecute the application.

State		Error information	Action
	Accessed during system failure	Empty string (*1)	
Switch to the standby server	Switched during access	57P01 (*1) Empty string (*1)	
	Accessed during switch	Empty string (*1)	

*1: This is the return value of the PostgresException attribute SqlState.

4.4.4 Notes

Type Plugins

- These type libraries include:
  - NodaTime - the recommended way to interact with PostgrSQL date/time types
  - Json.NET - allows Npgsql to use the NewtonSoft Json.NET library when reading and writing JSON data (both json and jsonb)
  - NetTopologySuite - allows Npgsql to map PostGIS spatial types directly to the NetTopology suite types (the leading spatial library in .NET)
  - GeoJSON - allows Npgsql to read and write PostGIS spatial types as GeoJSON types via the GeoJSON.NET library
- Setup the plugin in your application simply by adding a dependency on the plugin (this should have been done automatically when installed to the project) and set it up. See the following code snippet for an example of setting up the Npgsql.NodaTime plugin:

```
    using Npgsql;
    // Place this at the beginning of your program to use NodaTime everywhere
(recommended)
    NpgsqlConnection.GlobalTypeMapper.UseNodaTime();
    // Or to temporarily use NodaTime on a single connection only:
    conn.TypeMapper.UseNodaTime();
```

Once the plugin is setup, you can read and write NodaTime objects as per the code snippet below:

```
    // Write NodaTime Instant to PostgreSQL "timestamp without time zone"
    using (var cmd = new NpgsqlCommand(@"INSERT INTO mytable (my_timestamp) VALUES
(@p)", conn))
    {
        cmd.Parameters.Add(new NpgsqlParameter("p", Instant.FromUtc(2011, 1, 1, 10,
30)));
        cmd.ExecuteNonQuery();
    }

    // Read timestamp back from the database as an Instant
    using (var cmd = new NpgsqlCommand(@"SELECT my_timestamp FROM mytable", conn))
    using (var reader = cmd.ExecuteReader())
    {
        reader.Read();
        var instant = reader.GetFieldValue<Instant>(0);
    }
```

- To apply the type plugin updates, do one of the following:
  - After uninstalling the type plugin (Refer to "4.5.2 Uninstalling .NET Data Provider Type Plugins"), setup the type plugin (Refer to "4.2.2 Setting Up .NET Data Provider Type Plugins**").**

- Remove the type plugin directory from the packages directory of the solution, and then restore it using the nuget restore command.

- When you deploy an application with a type plugin, the type plugin is included in the distribution. Therefore, after applying the type plugin updates, you must rebuild the application and deploy the updated application.

Additional Notes on each Type Plugin

Describe notes about each type plugin.

NodaTime

Describes the mapping of PostgreSQL data types to NodaTime data types.

Mapping Table

PostgreSQL Type	Default NodaTime Type	Additional NodaTime Type	Note
timestamp with time zone	Instant	ZonedDateTime(*1), OffsetDateTime(*1)	UTC timestamp in database. Only UTC ZonedDateTime and OffsetDateTime are supported.
timestamp without time zone	LocalDateTime(*2)		A timestamp in an unknown or implied timezone.
date	LocalDate		A simple date with no timezone or offset information.
time without time zone	LocalTime		A simple time with no timezone or offset information.
time with time zone	OffsetTime		A type that stores a time and an offset. Generally not recommended for use.
interval	Period	Duration	A time interval from fractional seconds to years. NodaTime Duration is supported for days and smaller intervals, but not for years or months (because they have no absolute duration). Duration can be used in any interval unit.
tszrange	Interval	NpgsqlRange<Instant> ,etc	The time interval between two time instances (start and end).
strange	NpgsqlRange<LocalDateTime>		The time interval between two timestamps in unknown or implied timezones.

PostgreSQL Type	Default NodaTime Type	Additional NodaTime Type	Note
daterange	DateInterval	NpgsqlRange<LocalDate> ,etc	The interval between two dates.

*1: When Npgsql.EnableLegacyTimestampBehavior is enabled, writing or reading ZonedDateTime or OffsetDateTime is automatically converted to UTC.

*2: When Npgsql.EnableLegacyTimestampBehavior is enabled, timestamp without time zone is mapped to Instant instead of LocalDateTime by default.

Json.NET

Once the JSON plugin has been setup, users can transparently read and write CLR objects as JSON values and the plugin will automatically serialize/deserialize them

See the code snippet below:

```
// Write arbitrary CLR types as JSON
using (var cmd = new NpgsqlCommand(@"INSERT INTO mytable (my_json_column) VALUES (@p)",
conn))
{
    cmd.Parameters.Add(new NpgsqlParameter("p", NpgsqlDbType.Jsonb) { Value =
MyClrType });
    cmd.ExecuteNonQuery();
}

// Read arbitrary CLR types as JSON
using (var cmd = new NpgsqlCommand(@"SELECT my_json_column FROM mytable", conn))
using (var reader = cmd.ExecuteReader())
{
    reader.Read();
    var someValue = reader.GetFieldValue<MyClrType>(0);
}
```

NetTopologySuite (spatial)

By default the plugin handles only ordinates provided by the DefaultCoordinateSequenceFactory of GeometryServiceProvider.Instance. If GeometryServiceProvider is initialized automatically the X and Y ordinates are handled. To change the behavior specify the handleOrdinates parameter like in the following example:

```
conn.TypeMapper.UseNetTopologySuite(handleOrdinates: Ordinates.XYZ);
```

To process the M ordinate, you must initialize GeometryServiceProvider.Instance to a new NtsGeometryServices instance with coordinateSequenceFactory set to a DotSpatialAffineCoordinateSequenceFactory. Or you can specify the factory when calling UseNetTopologySuite.

```
// Place this at the beginning of your program to use the specified settings everywhere
(recommended)
GeometryServiceProvider.Instance = new NtsGeometryServices(
    new DotSpatialAffineCoordinateSequenceFactory(Ordinates.XYM),
    new PrecisionModel(PrecisionModels.Floating),
    -1);

// Or specify settings for Npgsql only
conn.TypeMapper.UseNetTopologySuite(
    new DotSpatialAffineCoordinateSequenceFactory(Ordinates.XYM));
```

Reading and Writing Geometry Values

When reading PostGIS values from the database, Npgsql will automatically return the appropriate NetTopologySuite types: Point, LineString, and so on. Npgsql will also automatically recognize NetTopologySuite's types in parameters, and will automatically send the corresponding PostGIS type to the database. The following code demonstrates a roundtrip of a NetTopologySuite Point to the database:

```
var point = new Point(new Coordinate(1d, 1d));
conn.ExecuteNonQuery("CREATE TEMP TABLE data (geom GEOMETRY)");
using (var cmd = new NpgsqlCommand("INSERT INTO data (geom) VALUES (@p)", conn))
{
    cmd.Parameters.AddWithValue("@p", point);
    cmd.ExecuteNonQuery();
}

using (var cmd = new NpgsqlCommand("SELECT geom FROM data", conn))
using (var reader = cmd.ExecuteReader())
{
    reader.Read();
    Assert.That(reader[0], Is.EqualTo(point));
}
```

You may also explicitly specify a parameter's type by setting NpgsqlDbType.Geometry.

Geography (geodetic) Support

PostGIS has two types:geometry (for Cartesian coordinates) and geography (for geodetic or spherical coordinates). You can read about the geometry/geography distinction in the PostGIS docs. In a nutshell, geography is much more accurate when doing calculations over long distances, but is more expensive computationally and supports only a small subset of the spatial operations supported by geometry.

Npgsql uses the same NetTopologySuite types to represent both geometry and geography - the Point type represents a point in either Cartesian or geodetic space. You usually don't need to worry about this distinction because PostgreSQL will usually cast types back and forth as needed. However, it's worth noting that Npgsql sends Cartesian geometry by default, because that's the usual requirement. You have the option of telling Npgsql to send geography instead by specifying NpgsqlDbType.Geography:

```
using (var cmd = new NpgsqlCommand("INSERT INTO data (geog) VALUES (@p)", conn))
{
    cmd.Parameters.AddWithValue("@p", NpgsqlDbType.Geography, point);
    cmd.ExecuteNonQuery();
}
```

If you prefer to use geography everywhere by default, you can also specify that when setting up the plugin:

```
NpgsqlConnection.GlobalTypeMapper.UseNetTopologySuite(geographyAsDefault: true);
```

GeoJSON (spatial)

Using the GeoJSON plugin is the same as the NetTopologuSuite.

Notes on the Query Builder

- Prefix named parameters with "@".
- Uppercase object names cannot be used, even when enclosed in double quotation marks.
  To use uppercase object names enclosed in double quotation marks, include them in SQL statements and enter these in the [Generate the SQL statements] window rather than in the Query Builder.
- SQL statements cannot be correctly generated if the SQL statement specified in Filter matches any of the conditions below:
  - It uses PostgreSQL intrinsic operators such as << or ::.

- It uses functions with keywords such as AS, FROM, IN, OVER.

  Example: extract(field from timestamp), RANK( ) OVER

- It uses functions with the same names as those prescribed in SQL conventions, but that require different arguments.

Notes on metadata

- The CommandBehavior.KeyInfo argument must be specified if executing ExecuteReader before obtaining metadata using GetSchemaTable.

```
NpgsqlDataReader  ndr=cmd.ExecuteReader(CommandBehavior.KeyInfo);
DataTable dt = dr.GetSchemaTable();
```

Notes on automatically generating update-type SQL statements

- If the SQL statement includes a query (which cannot be updated) that matches any of the conditions below, an update-type SQL statement will be generated (note that it may not be possible to execute this SQL statement in some cases):

  - It includes derived tables
  - It includes the same column name as the select list

  Update-type SQL statements will be automatically generated in the following cases:

  - If update statements are obtained using NpgsqlCommandBuilder
  - If data is updated using NpgsqlDataAdapter

Notes on distributed transactions

- Applications using transaction scope can use distributed transactions by linking with Microsoft Distributed Transaction Coordinator (MSDTC). In this case, note the following:

  - Ensure that the value of max_prepared_transactions is greater than max_connection, so that "PREPARE TRANSACTION" can be issued for each transaction that simultaneously connects to the database server.
  - If each transaction in the transaction scope accesses the same resource using different connections, the database server will perceive it as requests from different applications, and a deadlock may occur. By configuring a timeout value for the transaction scope beforehand, the deadlock can be broken.

4.5 Uninstallation

This section explains how to uninstall Npgsql and Npgsql for Entity Framework.

4.5.1 Uninstalling Npgsql

Npgsql is installed per project. To uninstall it, follow the procedure below:

1. In Visual Studio, open a project for which Npgsql is installed.
2. Click [Tools] >> [NuGet Package Manager] >> [Manage NuGet Packages for Solution].
3. Select all the projects that have Npgsql installed, and then click [Uninstall].
   Alternatively, if the plugin packages have been removed and is no longer installed, in the solution explorer open the Dependencies/NuGet node and delete the plugins that require uninstallation.

4.5.2 Uninstalling .NET Data Provider Type Plugins

The **.NET Data Provider** type plugins are installed per project. To uninstall it, follow the procedure below:

1. In Visual Studio, open a project for which Npgsql the Plugin to be removed is installed.
2. Click [Tools] >> [NuGet Package Manager] >> [Manage NuGet Packages for Solution].
3. Select all the projects that have Npgsql Plugin(s) installed, and then click [Uninstall].
   Alternatively, if the plugin packages have been removed and are no longer installed, in the solution explorer open the Dependencies/NuGet node and delete the plugins that require uninstallation.

4.5.3 Uninstalling Entity Framework Core

Entity Framework Core is installed per project. To uninstall it, follow the procedure below:

1. In Visual Studio, open a project for which Entity Framework Core is installed.
2. Click [Tools] >> [NuGet Package Manager] >> [Manage NuGet Packages for Solution].
3. Select all the projects that have Entity Framework Core installed, and then click [Uninstall].
   Alternatively, if the Entity Framework package has been removed and is no longer installed, in the solution explorer open the Dependencies/NuGet node and delete the packages that require uninstallation.

4.5.4 Uninstalling Npgsql.OpenTelemetry

Npgsql.OpenTelemetry is installed per project. To uninstall it, follow the procedure below:

1. In Visual Studio, open a project for which Npgsql.OpenTelemetry is installed.
2. Click [Tools] >> [NuGet Package Manager] >> [Manage NuGet Packages for Solution].
3. Select all the projects that have Npgsql.OpenTelemetry installed, and then click [Uninstall].
   Alternatively, if the Npgsql.OpenTelemetry package has been removed and is no longer installed, in the solution explorer open the Dependencies/NuGet node and delete the packages that require uninstallation.

Chapter 5 C Library (libpq)

This chapter describes how to use C libraries.

5.1 Development Environment

Install Fujitsu Enterprise Postgres Client Package for the architecture to be developed and executed.

See

Refer to Installation and Setup Guide for Client for information on the C compiler required for C application development.

5.2 Setup

This section describes the environment settings required to use C libraries and how to encrypt data for communication.

5.2.1 Environment Settings

To execute an application that uses libpq, set the environment variable as shown below.

L

Linux

- Required for execution of the application
  - PGLOCALEDIR
    fujitsuEnterprisePostgresClientInstallDir/share/locale

Example

"<*x*>" indicates the product version.

```
> PGLOCALEDIR=/opt/fsepv<x>client64/share/locale;export PGLOCALEDIR
```

W

Windows (R)

- Required for compile/link
  - LIB
    fujitsuEnterprisePostgresClientInstallDir\lib
  - INCLUDE
    fujitsuEnterprisePostgresClientInstallDir\include
- Required for execution of the application
  - PATH
    fujitsuEnterprisePostgresClientInstallDir\lib
  - PGLOCALEDIR
    fujitsuEnterprisePostgresClientInstallDir\share\locale

Example

When the 32-bit version client package is installed on a 64-bit operating system. "<*x*>" indicates the product version.

```
> SET PATH=%ProgramFiles(x86)%\Fujitsu\fsepv<x>client32\lib;%PATH%
> SET LIB=%ProgramFiles(x86)%\Fujitsu\fsepv<x>client32\lib;%LIB%
```

```
> SET INCLUDE=%ProgramFiles(x86)%\Fujitsu\fsepv<x>client32\include;%INCLUDE%
> SET PGLOCALEDIR=%ProgramFiles(x86)%\Fujitsu\fsepv<x>client32\share\locale
```

5.2.2 Message Language and Encoding System Used by Applications Settings

This section explains the language settings for the application runtime environment and the encoding settings for the application.

Language settings

You must match the language settings for the application runtime environment with the message locale settings of the database server.

Messages output by an application may include text from messages sent from the database server. In the resulting text, the text of the application message will use the message locale of the application, and the text of the message sent by the database server will use the message locale of the database server. If the message locales do not match, more than one language or encoding system will be used. Moreover, if the encoding systems do not match, characters in the resulting text can be garbled.

L

- Linux

Set the locale for messages (LC_MESSAGES category) to match the message locale of the database server. This can be done in a few different ways, such as using environment variables. Refer to the relevant manual of the operating system for information on the setlocale function.

Example

Example of specifying "en_US.UTF-8" with the setlocale function

```
setlocale(LC_ALL,"en_US.UTF-8");
```

Specifying the locale of the LC_ALL category propagates the setting to LC_MESSAGE.

W

- Windows(R)

Align the locale of the operating system with the message locale of the database server.

Encoding System Settings

Ensure that the encoding system that is embedded in the application and passed to the database, and the encoding system setting of the runtime environment, are the same. The encoding system cannot be converted correctly on the database server.

Use one of the following methods to set the encoding system for the application:

- Set the PGCLIENTENCODING environment variable in the runtime environment.
- Set the client_encoding keyword in the connection string.
- Use the PQsetClientEncoding function.

See

Refer to "Supported Character Sets" in "Server Administration" in the PostgreSQL Documentation for information on the strings that represent the encoding system that can be set.

For example, when using "Unicode" and "8 bit", set the string "UTF8".

Note

Text may be garbled when outputting results to the command prompt. Review the font settings for the command prompt if this occurs.

5.2.3 Settings for Encrypting Communication Data

Set in one of the following ways when performing remote access using communication data encryption:

When setting from outside with environment variables

Specify "require", "verify-ca", or "verify-full" in the PGSSLMODE environment variable.

In addition, the parameters for the PGSSLROOTCERT and PGSSLCRL environment variables need to be set to prevent spoofing of the database server.

See

Refer to "Environment Variables" in "Client Interfaces" in the PostgreSQL Documentation for information on environment variables.

When specifying in the connection URI

Specify "require", "verify-ca", or "verify-full" in the "sslmode" parameter of the connection URI.

In addition, the parameters for the sslcert, sslkey, sslrootcert, and sslcrl need to be set to prevent spoofing of the database server.

See

Refer to "Secure TCP/IP Connections with SSL" in "Server Administration" in the PostgreSQL Documentation for information on encrypting communication data.

5.3 Connecting with the Database

Point

Use the connection service file to specify the connection destination. In the connection service file, a name (service name) is defined as a set, comprising information such as connection destination information and various types of tuning information set for connections. By using the service name defined in the connection service file when connecting to databases, it is no longer necessary to modify applications when the connection information changes.

Refer to "Client Interfaces", "The Connection Service File" in the PostgreSQL Documentation for details.

See

Refer to "Database Connection Control Functions" in "Client Interfaces" in the PostgreSQL Documentation.

In addition, refer to "6.3 Connecting with the Database" in "Embedded SQL in C " for information on connection string.

5.4 Application Development

See

Refer to "libpq - C Library" in "Client Interfaces" in the PostgreSQL Documentation for information on developing applications.

However, if you are using the C library, there are the following differences to the PostgreSQL C library (libpq).

5.4.1 Compiling Applications

Specify the following paths when compiling applications.

Refer to your compiler documentation for information on how to specify the path.

Also, when using a shared library, refer to "A.6 How to Build and Run an Application that Uses Shared Libraries".

L - Linux

L Table 5.1 Include file and library path

Type of path	Path name
Path of the include file	*fujitsuEnterprisePostgresClientInstallDir*/include
Path of the library	*fujitsuEnterprisePostgresClientInstallDir*/lib

L Table 5.2 C Library (libpq library)

Type of library	Library name
Dynamic library	libpq.so
Static library	libpq.a

W - Windows(R)

If the include file and the library path have been set in the environment variable, there is no need to specify the paths shown below for the compile.

W Table 5.3 Include file and library path

Type of path	Path name
Path of the include file	*fujitsuEnterprisePostgresClientInstallDir*\include
Path of the library	*fujitsuEnterprisePostgresClientInstallDir*\lib

W Table 5.4 C Library (libpq library)

Type of library	Library name
Library for links	libpq.lib
Dynamic library	libpq.dll

5.4.2 Creating Applications while in Database Multiplexing Mode

This section explains points to consider when creating applications while in database multiplexing mode.

See

- Refer to the Cluster Operation Guide (Database Multiplexing) for information on database multiplexing mode.
- Refer to "Application Development" in the Cluster Operation Guide (PRIMECLUSTER) for points to consider when creating applications using the failover feature integrated with the cluster software.

5.4.2.1 Errors when an Application Connection Switch Occurs and Corresponding Actions

If an application connection switch occurs while in database multiplexing mode, explicitly close the connection and then reestablish the connection or reexecute the application.

The table below shows errors that may occur during a switch, and the corresponding action to take.

State		Error information	Action
Server failure or Fujitsu Enterprise Postgres system failure	Failure occurs during access	PGRES_FATAL_ERROR(*1) 57P01(*2) NULL(*2)	After the switch is complete, reestablish the connection, or reexecute the application.
	Accessed during system failure	CONNECTION_BAD(*3)	
Switch to the standby server	Switched during access	PGRES_FATAL_ERROR(*1) 57P01(*2) NULL(*2)	
	Accessed during switch	CONNECTION_BAD(*3)	

*1: Return value of PQresultStatus().

*2: Return value of PQresultErrorField() PG_DIAG_SQLSTATE.

*3: Return value of PQstatus().

Chapter 6 Embedded SQL in C

This chapter describes application development using embedded SQL in C.

6.1 Development Environment

Install Fujitsu Enterprise Postgres Client Package for the architecture to be developed and executed.

See

Refer to Installation and Setup Guide for Client for information on the C compiler required for C application development.

Note

C++ is not supported. Create a library by implementing embedded SQL in C, and call it from C++.

6.2 Setup

6.2.1 Environment Settings

When using embedded SQL in C, the same environment settings as when using the C library (libpq) are required.

Refer to "5.2.1 Environment Settings" in "C Library (libpq)" for information on the environment settings for the library for C.

Additionally, set the following path for the precompiler ecpg in the PATH environment variable:

L Linux

```
fujitsuEnterprisePostgresClientInstallDir/bin
```

W Windows(R)

```
fujitsuEnterprisePostgresClientInstallDir\bin
```

6.2.2 Message Language and Encoding System Used by Applications Settings

The message language and the encoding System Settings Used by Applications settings are the same as when using the library for C.

However, in embedded SQL, the PQsetClientEncoding function cannot be used in the encoding system settings. In embedded SQL, use the SET command to specify the encoding system in client_encoding.

Refer to "5.2.2 Message Language and Encoding System Used by Applications Settings" in "C Library (libpq)" for information on the settings for the library for C.

6.2.3 Settings for Encrypting Communication Data

When encrypting the communication data, the same environment settings as when using the C library (libpq) are required.

Refer to "5.2.3 Settings for Encrypting Communication Data" in "C Library (libpq)" for information on the environment settings for the C library.

6.3 Connecting with the Database

Point

- It is recommended to use a connection service file to specify connection destinations. In the connection service file, a name (service name) is defined as a set, comprising information such as connection destination information and various types of tuning information set for connections. By using the service name defined in the connection service file when connecting to databases, it is no longer necessary to modify applications when the connection information changes. Refer to "The Connection Service File" in "Client Interfaces" in the PostgreSQL Documentation for information.
- If using a connection service file, perform either of the procedures below:
  - Set the service name as a string literal or host variable, as follows:

    tcp:postgresql://?service=my_service
  - Set the service name in the environment variable PGSERVICE, and use CONNECT TO DEFAULT

Use the CONNECT statement shown below to create a connection to the database server.

Format

```
EXEC SQL CONNECT TO target [AS connection-name] [USER user-name];
```

target

Write in one of the following formats:

- dbname@host:port
- tcp:postgresql://host:port/dbname[?options]
- unix:postgresql://host[:port][/dbname][?options]
  (Definition method when using the UNIX domain socket)
- SQL string literal containing one of the above formats
- Reference to a character variable containing one of the above formats
- DEFAULT

user-name

Write in one of the following formats:

- username
- username/password
- username IDENTIFIED BY password
- username USING password

Description of the arguments

Argument	Description
dbname	Specify the database name.
host	Specify the host name for the connection destination.
port	Specify the port number for the database server. The default is "27500".
connection-name	Specify connection names to identify connections when multiple connections are to be processed within a single program.

Argument	Description
username	Specify the user that will connect with the database. If this is omitted, the name used will be that of the user on the operating system that is executing the application.
password	Specify a password when authentication is required.
options	Specify the following parameter when specifying a time for timeout. Connect parameters with & when specifying more than one. The following shows the values specified for each parameter. - connect_timeout Specify the timeout for connections. Specify a value between 0 and 2147483647 (in seconds). There is no limit set if you set 0 or an invalid value. If "1" is specified, the behavior will be the same as when "2" was specified. An error occurs when a connection cannot be established within the specified time. - keepalives This enables keepalive. Keepalive is disabled if 0 is specified. Keepalive is enabling when any other value is specified. The default is keepalive enabled. Keepalive causes an error to occur when it is determined that the connection with the database is disabled. - keepalives_idle Specify the time until the system starts sending keepalive messages when communication with the database is not being performed. - Linux Specify a value between 1 and 32767 (in seconds). The default value of the system is used if this is not specified. - Windows(R) Specify a value between 1 and 2147483647 (in seconds). 7200 will be set as default if a value outside this range is specified or if nothing is specified. - keepalives_interval Specify the interval between resends when there is no response to keepalive messages. - Linux Specify a value between 1 and 32767 (in seconds). The default value of the system is used if this is not specified. - Windows(R) Specify a value between 1 and 2147483647 (in seconds). 1 will be set as default if a value outside this range is specified or if nothing is specified. - keepalives_count Specify the number of resends for keepalive messages. - Linux Specify a value between 1 and 127. The default value of the system is used if this is not specified. - Windows(R) The system default value is used irrespective of what is specified for this parameter. - tcp_user_timeout

Argument	Description
	After establishing the connection, when sending from the client to the server, if the TCP resend process operates, specify the time until it is considered to be disconnected. - Linux Specify a value between 0 and 2147483647 (in millseconds). The default value of the system is used if 0. 0 will be set as default if nothing is specified. - Windows(R) Cannot be specified.

L

Note

If a value other than 0 is specified for the tcp_user_timeout parameter, the waiting time set by the tcp_keepalives_idle parameter and tcp_keepalives_interval parameter will be invalid and the waiting time specified by the tcp_user_timeout parameter will be used.

Code examples for applications

```
EXEC SQL CONNECT TO tcp:postgresql://sv1:27500/mydb?
connect_timeout=20&keepalives_idle=20&keepalives_interval=5&keepalives_count=2&keepalives=
1 USER myuser/myuser01;
```

6.4 Application Development

Refer to "ECPG - Embedded SQL in C" in "Client Interfaces" in the PostgreSQL Documentation for information on developing applications.

However, when using embedded SQL in C, there are the following differences to the embedded SQL (ECPG) in PostgreSQL C.

6.4.1 Support for National Character Data Types

This section describes how to use the national character data types using the SQL embedded C preprocessor.

The following explains the C language variable types corresponding to the NCHAR type:

Specify the number of characters specified for the NCHAR type multiple by 4, plus 1 for the length of the host variable.

Data Type	Host variable type
NATIONAL CHARACTER(*n*)	NCHAR variable name [nx4+1]
NATIONAL CHARACTER VARYING(*n*)	NVARCHAR variable name [nx4+1]

See

Refer to "Handling Character Strings" in "Client Interfaces" in the PostgreSQL documentation for information on using character string types.

6.4.2 Compiling Applications

Append the extension "pgc" to the name of the source file for the embedded SQL in C.

When the pgc file is precompiled using the ecpg command, C source files will be created, so use the C compiler for the compile.

Precompiling example

```
ecpg testproc.pgc
```

If an optimizer hint block comment is specified for the SQL statement, specify the following option in the ecpg command:

--enable-hint

Enables the optimizer hint block comment (hereafter, referred to as the "hint clause"). If this option is not specified, the hint clause will be removed as a result of the ecpg precompile and be disabled.

The SQL statements that can be specified in the hint clause are SELECT, INSERT, UPDATE, and DELETE.

The locations in which the hint clause can be specified are immediately after one of the SELECT, INSERT, UPDATE, DELETE, or WITH keywords. A syntax error will occur if any other location is specified.

Example of specifying the hint clause

```
EXEC SQL SELECT /*+ IndexScan(prod ix01) */ name_id INTO :name_id FROM prod WHERE id
= 1;
```

Refer to "11.1.1 Optimizer Hints" for information on optimizer hints.

Note

Take the following points into account when using embedded SQL source files:

- Multibyte codes expressed in SJIS or UTF-16 cannot be included in statements or host variable declarations specified in EXEC SQL.
- Do not use UTF-8 with a byte order mark (BOM), because an error may occur during compilation if the BOM character is incorrectly recognized as the source code.
- Multibyte characters cannot be used in host variable names.
- It is not possible to use a TYPE name that contains multibyte characters, even though it can be defined.

Specify the following paths when compiling a C application output with precompiling.

Refer to your compiler documentation for information on how to specify the path.

Also, when using a shared library, refer to "A.6 How to Build and Run an Application that Uses Shared Libraries".

L Linux

L Table 6.1 Include file and library path

Type of path	Path name
Path of the include file	*fujitsuEnterprisePostgresClientInstallDir*\include
Path of the library	*fujitsuEnterprisePostgresClientInstallDir*\lib

L Table 6.2 C Library

Type of library	Library name	Note
Dynamic library	libecpg.so	
	libpgtypes.so	When using the pgtypes library
Static library	libecpg.a	
	libpgtypes.a	When using the pgtypes library

Windows(R)

If the include file and the library path have been set in the environment variable, there is no need to specify the paths shown below for the compile.

Table 6.3 Include file and library path

Type of path	Path name
Path of the include file	*fujitsuEnterprisePostgresClientInstallDir\include*
Path of the library	*fujitsuEnterprisePostgresClientInstallDir\lib*

Table 6.4 C Library

Type of library	Library name	Note
Library for links	libecpg.lib	
	libpgtypes.lib	When using the pgtypes library
Dynamic library	libecpg.dll	
	libpgtypes.dll	When using the pgtypes library

Note

- The libecpg library in Windows(R) is created by "release" and "multithreaded" options. When using the ECPGdebug function included in this library, compile using the "release" and "multithreaded" flags in all programs that use this library. When you do this, use the "dynamic" flag if you are using libecpg.dll, and use the "static" flag if you are using libecpg.lib.

  Refer to "Library Functions" in "Client Interfaces" in the PostgreSQL Documentation for information on the ECPGdebug function.

- The cl command expects input to be a program that uses one of the following code pages, so convert the program to these code pages and then compile and link it (refer to the Microsoft documentation for details).

  - ANSI console code pages (example: Shift-JIS for Japanese)
  - UTF-16 little-endian with or without BOM (Byte Order Mark)
  - UTF-16 big-endian with or without BOM
  - UTF-8 with BOM

  The cl command converts strings in a program to an ANSI console code page before generating a module, so the data sent to and received from the database server becomes an ANSI console code page. Therefore, set the coding system corresponding to the ANSI console code page as the coding system of the client.

  Refer to "Character Set Support" in "Server Administration" in the PostgreSQL Documentation for information on how to set the client encoding system.
  (Example: To use environment variables in Japanese, set SJIS in PGCLIENTENCODING.)

6.4.3 Bulk INSERT

This section describes the bulk INSERT.

Synopsis

```
EXEC SQL [ AT conn ] [ FOR { numOfRows | ARRAY_SIZE } ]
    INSERT INTO tableName [ ( colName [, ...] ) ]
    { VALUES ( { expr | DEFAULT } [, ...] ) [, ...] | query }
```

```
[ RETURNING * | outputExpr [ [ AS ] outputName ] [, ...]
INTO outputHostVar [ [ INDICATOR ] indicatorVar  ]  [, ...] ];
```

Description

Bulk INSERT is a feature that inserts multiple rows of data in bulk.

By specifying the array host variable that stored the data in the VALUES clause of the INSERT statement, the data for each element in the array can be inserted in bulk. This feature is used by specifying the insertion count in the FOR clause immediately before the INSERT statement.

FOR Clause

Specify the insertion count using numOfRows or ARRAY_SIZE in the FOR clause. The FOR clause can be specified only in the INSERT statement, not in other update statements.

numOfRows and ARRAY_SIZE

Insertion processing will be executed only for the specified count. However, if the count is 1, it will be assumed that the FOR clause was omitted when the application is executed. In this case, proceed according to the INSERT specification in the PostgreSQL Documentation.

Specify the FOR clause as an integer host variable or as a literal.

Specify ARRAY_SIZE to insert all elements of the array in the table. When specifying ARRAY_SIZE, specify at least one array in *expr*.

If two or more arrays were specified in expr, it will be assumed that ARRAY_SIZE is the minimum number of elements in the array.

numOfRows or ARRAY_SIZE must exceed the minimum number of elements in all arrays specified in *expr*, *outputHostVar*, and *indicatorVal*.

The following example shows how to specify the FOR clause.

```
int  number_of_rows = 10;
int id[25];
char name[25][10];

EXEC SQL FOR :number_of_rows    /* will process 10 rows */
INSERT INTO prod (name, id) VALUES (:name, :id);

EXEC SQL FOR ARRAY_SIZE           /* will process 25 rows */
INSERT INTO prod (name, id) VALUES (:name, :id);
```

expr

Specify the value to be inserted in the table. Array host variables, host variable literals, strings, and pointer variables can be specified. Structure type arrays and pointer variable arrays cannot be specified.

Do not use pointer variables and ARRAY_SIZE at the same time. The reason for this is that the number of elements in the area represented by the pointer variable cannot be determined.

query

A query (SELECT statement) that supplies the rows to be inserted. The number of rows returned by *query* must be 1. If two or more rows are returned, an error will occur. This cannot be used at the same time as ARRAY_SIZE.

outputHostVar, indicatorVal

These must be array host variables or pointer variables.

Error Messages

Given below are the error messages that are output when bulk INSERT functionality is not used correctly.

Message

invalid statement name "FOR value should be positive integer"

Cause

The value given for *numOfRows* is less than or equal to 0.

Solution

Specify a value that is more than or equal to 1 for *numOfRows*.

Message

invalid statement name "Host array variable is needed when using FOR ARRAY_SIZE"

Cause

A host array is not specified in the values clause when using the ARRAY_SIZE keyword.

Solution

At least one host array variable should be included in the values clause

Message

SELECT...INTO returns too many rows

Cause

The number of rows returned by the 'SELECT ... INTO' query in the INSERT statement is more than one.

Solution

When the value of *numOfRows* is more than one, the maximum number of rows that can be returned by the 'SELECT ... INTO' query in the INSERT statement is one.

Limitations

The limitations when using bulk INSERT are given below.

- Array of structures should not be used as an input in the 'VALUES' clause. Attempted use will result in junk data being inserted into the table.
- Array of pointers should not be used as an input in the 'VALUES' clause. Attempted use will result in junk data being inserted into the table.
- ECPG supports the use of 'WITH' clause in single INSERT statements. 'WITH' clause cannot be used in bulk INSERT statements.
- ECPG does not calculate the size of the pointer variable. So when a pointer variable is used that includes multiple elements, *numOfRows* should be less than or equal to the number of elements in the pointer. Otherwise, junk data will be inserted into the table.
- If an error occurs, all bulk INSERT actions will be rolled back, therefore, no rows are inserted. However, if the RETURNING clause was used, and the error occurred while obtaining the rows after the insertion was successful, the insertion processing will not be rolled back.

Samples

Given below are some sample usages of the bulk INSERT functionality.

Basic Bulk INSERT

```
int in_f1[4] = {1,2,3,4};
...
EXEC SQL FOR 3 INSERT INTO target (f1) VALUES (:in_f1);
```

The number of rows to insert indicated by the FOR clause is 3, so the data in the first 3 elements of the host array variable are inserted into the table. The contents of the target table will be:

```
 f1
----
  1
  2
  3
(3 rows)
```

Also a host integer variable can be used to indicate the number of rows that will be inserted in FOR clause, which will produce the same result as above:

```
int num = 3;
int in_f1[4] = {1,2,3,4};
...
EXEC SQL FOR :num INSERT INTO target (f1) VALUES (:in_f1);
```

Inserting constant values

Constant values can also be bulk INSERTed into the table as follows:

```
EXEC SQL FOR 3 INSERT INTO target (f1,f2) VALUES (DEFAULT,'hello');
```

Assuming the 'DEFAULT' value for the 'f1' column is '0', the contents of the target table will be:

```
f1 |  f2
---+-------
 0 | hello
 0 | hello
 0 | hello
(3 rows)
```

Using ARRAY_SIZE

'FOR ARRAY_SIZE' can be used to insert the entire contents of a host array variable, without explicitly specifying the size, into the table.

```
    int in_f1[4] = {1,2,3,4};
...
EXEC SQL FOR ARRAY_SIZE INSERT INTO target (f1) VALUES (:in_f1);
```

In the above example, four rows are inserted into the table.

If there are multiple host array variables specified as input values, then the number of rows inserted is same as the smallest array size. The example given below demonstrates this usage.

```
int in_f1[4] = {1,2,3,4};
char in_f3[3][10] = {"one", "two", "three"};
...
EXEC SQL FOR ARRAY_SIZE INSERT INTO target (f1,f3) VALUES (:in_f1,:in_f3);
```

In the above example, the array sizes are 3 and 4. Given that the smallest array size is 3, only three rows are inserted into the table. The table contents are given below.

```
 f1 |  f3
----+-------
  1 | one
  2 | two
```

```
  3 | three
(3 rows)
```

Using Pointers as Input

Pointers that contain multiple elements can be used in bulk INSERT.

```
int *in_pf1 = NULL;
in_pf1 = (int*)malloc(4*sizeof(int));
in_pf1[0]=1;
in_pf1[1]=2;
in_pf1[2]=3;
in_pf1[3]=4;
...
EXEC SQL FOR 4 INSERT INTO target (f1) values (:in_pf1);
```

The above example will insert four rows into the target table.

Using SELECT query

When using bulk INSERT, the input values can be got from the results of a SELECT statement. For example,

```
EXEC SQL FOR 4 INSERT INTO target(f1) SELECT age FROM source WHERE name LIKE 'foo';
```

Assuming that the 'SELECT' query returns one row, the same row will be inserted into the target table four times.

If the 'SELECT' query returns more than one row, the INSERT statement will throw an error.

```
EXEC SQL FOR 1 INSERT INTO target(f1) SELECT age FROM source;
```

In the above example, all the rows returned by the 'SELECT' statement will be inserted into the table. In this context '1' has the meaning of 'returned row equivalent'.

Using RETURNING clause

Bulk INSERT supports the same RETURNING clause syntax as normal INSERT. An example is given below.

```
int out_f1[4];
int in_f1[4] = {1,2,3,4};
...
EXEC SQL FOR 3 INSERT INTO target (f1) VALUES (:in_f1) RETURNING f1 INTO :out_f1;
```

After the execution of the above INSERT statement, the 'out_f1' array will have 3 elements with the values of '1','2' and '3'.

6.4.4 Creating Applications while in Database Multiplexing Mode

This section explains points to consider when creating applications while in database multiplexing mode.

- Refer to the Cluster Operation Guide (Database Multiplexing) for information on database multiplexing mode.
- Refer to "Application Development" in the Cluster Operation Guide (PRIMECLUSTER) for points to consider when creating applications using the failover feature integrated with the cluster software.

6.4.4.1 Errors when an Application Connection Switch Occurs and Corresponding Actions

If an application connection switch occurs while in database multiplexing mode, explicitly close the connection and then reestablish the connection or reexecute the application.

The table below shows errors that may occur during a switch, and the corresponding action to take.

State		Error information (*1)	Action
Server failure or Fujitsu Enterprise Postgres system failure	Failure occurs during access	57P01 57P02 YE000 26000 40001	After the switch is complete, reestablish the connection, or reexecute the application.
	Accessed during node/system failure	08001	
Switch to the standby server	Switched during access	57P01 57P02 YE000 26000 40001	
	Accessed during switch	08001	

*1: Return value of SQLSTATE.

6.4.5 Notes

Notes on creating multithreaded applications

In embedded SQL in C, DISCONNECT ALL disconnects all connections within a process, and therefore it is not thread-safe in all operations that use connections. Do not use it in multithreaded applications.

Chapter 7 Embedded SQL in COBOL

This chapter describes application development using embedded SQL in COBOL.

7.1 Development Environment

Install Fujitsu Enterprise Postgres Client Package for the architecture to be developed and executed.

See

Refer to the Installation and Setup Guide for Client for information on the COBOL compiler required for COBOL application development.

7.2 Setup

7.2.1 Environment Settings

When using embedded SQL in COBOL, the same environment settings as when using the C library (libpq) are required. Refer to "5.2.1 Environment Settings" in "C Library (libpq)" for information on the environment settings for the library for C.

Additionally, set the following path for the precompiler ecobpg in the PATH environment variable:

L

Linux

```
fujitsuEnterprisePostgresClientInstallDir/bin
```

W

Windows(R)

```
fujitsuEnterprisePostgresClientInstallDir\bin
```

7.2.2 Message Language and Encoding System Used by Applications

The settings for the message language and the encoding system used by applications should be the same as those required when using the library for C.

However, in embedded SQL, the PQsetClientEncoding function cannot be used in the encoding system settings. In embedded SQL, use the SET command to specify the encoding system in client_encoding.

Refer to "5.2.2 Message Language and Encoding System Used by Applications Settings" in "C Library (libpq)" for information on the settings for the library for C.

7.2.3 Settings for Encrypting Communication Data

When encrypting the communication data, the same environment settings as when using the C library (libpq) are required.

Refer to "5.2.3 Settings for Encrypting Communication Data" in "C Library (libpq)" for information on the environment settings for the C library.

7.3 Connecting with the Database

Use the CONNECT statement shown below to create a connection to the database server.

Format

```
EXEC SQL CONNECT TO target [AS connection-name] [USER user-name]END-EXEC.
```

target

Write in one of the following formats:

- dbname@host:port
- tcp:postgresql://host:port/dbname[?options]
- unix:postgresql://host[:port][/dbname][?options]
  (Definition method when using the UNIX domain socket)
- SQL string literal containing one of the above formats
- Reference to a character variable containing one of the above formats
- DEFAULT

Note

If target is DEFAULT, the AS clause and USER clause cannot be specified.

user-name

Write in one of the following formats:

- username
- username/password
- username IDENTIFIED BY password
- username USING password

Description of the arguments

Argument	Description
dbname	Specify the database name.
host	Specify the host name for the connection destination.
port	Specify the port number for the database server. The default is "27500".
connection-name	Specify connection names to identify connections when multiple connections are to be processed within a single program.
username	Specify the user that will connect with the database. If this is omitted, the name used will be that of the user on the operating system that is executing the application.
password	Specify a password when authentication is required.
options	Specify the following parameter when specifying a time for timeout. Connect parameters with & when specifying more than one. The following shows the values specified for each parameter. - connect_timeout Specify the timeout for connections. Specify a value between 0 and 2147483647 (in seconds). There is no limit set if you set 0 or an invalid value. If "1" is specified, the behavior will be the same as when "2" was specified. An error occurs when a connection cannot be established within the specified time. - keepalives This enables keepalive.

Argument	Description
	Keepalive is disabled if 0 is specified. Keepalive is enabled when any other value is specified. The default is keepalive enabled. Keepalive causes an error to occur when it is determined that the connection with the database is disabled. - keepalives_idle Specify the time until the system starts sending keepalive messages when communication with the database is not being performed. - Linux Specify a value between 1 and 32767 (in seconds). The default value of the system is used if this is not specified. - Windows(R) Specify a value between 1 and 2147483647 (in seconds). 7200 will be set as default if a value outside this range is specified or if nothing is specified. - keepalives_interval Specify the interval between resends when there is no response to keepalive messages. - Linux Specify a value between 1 and 32767 (in seconds). The default value of the system is used if this is not specified. - Windows(R) Specify a value between 1 and 2147483647 (in seconds). 1 will be set as default if a value outside this range is specified or if nothing is specified. - keepalives_count Specify the number of resends for keepalive messages. - Linux Specify a value between 1 and 127. The default value of the system is used if this is not specified. - Windows(R) The system default value is used irrespective of what is specified for this parameter.

Code examples for applications

```
EXEC SQL CONNECT TO tcp:postgresql://sv1:27500/mydb?
connect_timeout=20&keepalives_idle=20&keepalives_interval=5&keepalives_count=2&keepalives=
1 USER myuser/myuser01 END-EXEC.
```

7.4 Application Development

Refer to "Appendix D ECOBPG - Embedded SQL in COBOL" for information on developing applications.

7.4.1 Support for National Character Data Types

This section describes how to use the national character data types using the SQL embedded COBOL preprocessor.

The table below lists the COBOL variable types supporting the national character data types. The number of characters specified for the national character data type must be specified for the length of the host variable.

National character data type	COBOL variable type
CHARACTER(*n*)	*varName* PIC N(*n*)

National character data type	COBOL variable type
NATIONAL CHARACTER(*n*)	
CHARACTER VARYING(*n*) NATIONAL CHARACTER VARYING(*n*)	*varName* PIC N(*n*) VARYING

To use COBOL variable types that support national character data types, it is necessary to specify the ECOBPG_NCHAR environment variable.

ECOBPG_NCHAR={ UTF16LE | UTF16BE | UTF32LE | UTF32BE | SJIS }

In SQL embedded COBOL, specify the encoding of the COBOL variable types that support national character data types.

- UTF16LE: UTF-16 little-endian
- UTF16BE: UTF-16 big-endian
- UTF32LE: UTF-32 little-endian
- UTF32BE: UTF-32 big-endian
- SJIS: Shift JIS

If this environment variable is omitted, the encoding will be determined according to the encoding system of the client.

- If UTF8 is used: UTF16 (endians will be encoded in accordance with endians of the client system)
- If SJIS is used: SJIS

If encoding is specified for the translation option when compiling with NetCOBOL, the encoding specified for the national character data types should be used for the environment variable ECOBPG_NCHAR.

The list below shows NetCOBOL translation options and their corresponding environment variable ECOBPG_NCHAR values.

NetCOBOL translation options	Environment variable ECOBPG_NCHAR
ENCODE (UTF-8,UTF16,LE) RCS (UTF-16,LE)	UTF-16LE
ENCODE (UTF-8,UTF-16,BE) RCS (UTF-16,BE)	UTF-16BE
ENCODE (UTF-8,UTF-32,LE)	UTF-32LE
ENCODE (UTF-8,UTF-32,BE)	UTF-32BE
ENCODE (SJIS,SJIS)	SJIS
Not specified	No need to specify

Also, if the post-compiling encoding for an application differs from the locale of the execution environment, then the client encoding must be used for the application.

The list below shows the values supported for the combinations of application encoding, locale of the execution environment, and client encodings.

Application encoding	Locale used when executing an application	Client encoding
UTF-8	UTF-8	UTF-8
	SJIS	UTF-8
SJIS	UTF-8	SJIS

Application encoding	Locale used when executing an application	Client encoding
	SJIS	SJIS

Refer to "7.2.2 Message Language and Encoding System Used by Applications" for information on how to set client encoding systems.

The following example shows host variable declaration of a national character data type.

```
01 DATA1 PIC N(10).
01 DATA2 PIC N(10) VARYING.
```

Note

- Halfwidth characters should not be used for the national character data type COBOL variable.
- The national character data type column attribute obtained by applications should be the CHAR type.
- Encoding cannot be specified using the ENCODING clause, which is a feature of NetCOBOL.

7.4.2 Compiling Applications

Append the extension "pco" to the name of the source file for the embedded SQL in COBOL.

When the pco file is precompiled using the ecobpg command, COBOL source files will be created, so use the COBOL compiler for the compile.

Precompiling example

```
ecobpg testproc.pco
```

For COBOL code notation, "fixed" or "variable" format can be specified as an ecobpg command option. If not specified, "fixed" format is used.

Refer to "D.1 Precautions when Using Functions and Operators" and "D.12.1 ecobpg" for information on COBOL code notation and how to specify options for ecobpg.

If an optimizer hint block comment is specified for the SQL statement, specify the following option in the ecobpg command:

--enable-hint

Enables the optimizer hint block comment (hereafter, referred to as the "hint clause"). If this option is not specified, the hint clause will be removed as a result of the ecobpg precompile and be disabled.

The SQL statements that can be specified in the hint clause are SELECT, INSERT, UPDATE, and DELETE.

The locations in which the hint clause can be specified are immediately after one of the SELECT, INSERT, UPDATE, DELETE, or WITH keywords. A syntax error will occur if it is specified in any other location.

Example of specifying the hint clause

```
EXEC SQL SELECT /*+ IndexScan(prod ix01) */ name_id
INTO :name_id FROM prod WHERE id = 1 END-EXEC.
```

Refer to "11.1.1 Optimizer Hints" for information on optimizer hints.

If the encoding used for embedded SQL source files differs from that of the locale when precompiling was executed, set the encoding for the embedded SQL source files by specifying the following option for ecobpg.

-E-encode

Specify "UTF8", "SJIS", or "EUC_JP".

If this option is omitted, the encoding is processed based on the locale.

Path of the library file

The ecobpg command defines a group item "sqlca_t" to handle errors, which is defined in the library file stored in the following path:

Linux

Library file name	The storage destination of library file
SQLCA-COBOL.cob	*fujitsuEnterprisePostgresClientInstallDir*\include

Windows(R)

Library file name	The storage destination of library file
SQLCA-COBOL.cob	*fujitsuEnterprisePostgresClientInstallDir*\include

When the ecobpg command generates a COBOL file, it inserts a COPY statement with no options to copy the library file. Therefore, specify the path of the storage destination of library file when compiling. How to specify the path must conform to your compiler's specifications.

There is also a library file with the same contents without the extension"cob".

Information

Refer "D.7.2 sqlca"for information on the sqlca_t.

Libraries to use

The applications generated by ecobpg connect to PostgreSQL through the ECPG library. The ECPG library internally loads the libpq library.

Path of library

Refer to "6.4.2 Compiling Applications" for information on the location and name of the ECPG library. And Refer to " Chapter 5 C Library (libpq)" for information on the location and name of the libpq library.

Also, when using a shared library, refer to "A.6 How to Build and Run an Application that Uses Shared Libraries".

The COBOL compiler provides the how to link various libraries, so be sure to specify the path and libraries according to the specifications of your compiler.

Entry information of subprogram

If you use the ECPG library with a dynamic program structure, copy the entry information stored below. For details, follow the specifications of your compiler.

Linux

```
fujitsuEnterprisePostgresClientInstallDir/share/cobol_entry.info
```

Windows(R)

```
fujitsuEnterprisePostgresClientInstallDir\share\cobol_entry.info
```

Example

The examples of compiling the applications that dynamically links the COBOL language library.
Note that "<*x*>" indicates the product version.

Linux

- Linux 64-bit application:

```
cobol -M  -o testproc -I/opt/fsepv<x>client64/include -L/opt/fsepv<x>client64/lib -
lecpg -lpq testproc.cob
```

- Linux 64-bit application(When to set DT_RUNPATH):

```
cobol -c -M -o testproc.o -I/opt/fsepv<x>client64/include testproc.cob
ld -dynamic-linker /lib64/ld-linux-x86-64.so.2 -rpath /opt/fsepv<x>client64/lib --
enable-new-dtags -o testproc /usr/lib64/crti.o /usr/lib64/crt1.o /usr/lib64/crtn.o
testproc.o  -L/opt/fsepv<x>client64/lib -lecpg -lpq -Bdynamic -L/opt/FJSVcbl64/lib -
lrcobol -ldl -lc
```

Windows(R)

The examples of compiling on a 64-bit operating system.

- 64-bit application

```
> SET LIB=%ProgramFiles%\Fujitsu\fsepv<x>client64\lib;%LIB%
> cobol -I "%ProgramFiles%\Fujitsu\fsepv<x>client64\include" -M testproc.cob
> link testproc.obj F4AGCIMP.LIB LIBCMT.LIB LIBECPG.LIB LIBPQ.LIB /OUT:testproc.exe
```

- 32-bit application

[NetCOBOL V10.5 or earlier]

```
> SET LIB=%ProgramFiles(x86)%\Fujitsu\fsepv<x>client32\lib;%LIB%
> cobol32 -I "%ProgramFiles(x86)%\Fujitsu\fsepv<x>client32\include" -M testproc.cob
> link testproc.obj LIBC.LIB F3BICIMP.LIB LIBECPG.LIB LIBPQ.LIB /OUT:testproc.exe
```

[NetCOBOL V11.0 or later]

```
> SET LIB=%ProgramFiles(x86)%\Fujitsu\fsepv<x>client32\lib;%LIB%
> cobol32 -I "%ProgramFiles(x86)%\Fujitsu\fsepv<x>client32\include" -M testproc.cob
> link testproc.obj MSVCRT.LIB F3BICIMP.LIB LIBECPG.LIB LIBPQ.LIB /OUT:testproc.exe
```

7.4.3 Bulk INSERT

This section describes the bulk INSERT.

Synopsis

```
EXEC SQL [ AT conn ] [ FOR { numOfRows | ARRAY_SIZE } ]
    INSERT INTO tableName [ ( colName [, ...] ) ]
    { VALUES ( { expr | DEFAULT } [, ...] ) [, ...] | query }
    [ RETURNING * | outputExpr [ [ AS ] outputName ] [, ...]
    INTO outputHostVar [ [ INDICATOR ] indicatorVar ] [, ...] ] END-EXEC
```

Description

Bulk INSERT is a feature that inserts multiple rows of data in bulk.

By specifying the array host variable that stored the data in the VALUES clause of the INSERT statement, the data for each element in the array can be inserted in bulk. This feature is used by specifying the insertion count in the FOR clause immediately before the INSERT statement.

FOR Clause

Specify the insertion count using *numOfRows* or ARRAY_SIZE in the FOR clause. The FOR clause can be specified only in the INSERT statement, not in other update statements.

numOfRows and ARRAY_SIZE

Insertion processing will be executed only for the specified count. However, if the count is 1, it will be assumed that the FOR clause was omitted when the application is executed. In this case, proceed according to the INSERT specification in the PostgreSQL Documentation.

Specify the FOR clause as an integer host variable or as a literal.

Specify ARRAY_SIZE to insert all elements of the array in the table. When specifying ARRAY_SIZE, specify at least one array in *expr*.

If two or more arrays were specified in *expr*, it will be assumed that ARRAY_SIZE is the minimum number of elements in the array.

numOfRows or ARRAY_SIZE must exceed the minimum number of elements in all arrays specified in *expr*, *outputHostVar*, and *indicatorVal*.

The following example shows how to specify the FOR clause.

```
01 NUMBER-OF-ROWS PIC S9(9) COMP VALUE 10.
01 GROUP-ITEM.
05 ID1 PIC S9(9) OCCURS 25.
05 NAME PIC X(10) OCCURS 25.
* will process 10 rows
EXEC SQL FOR :NUMBER-OF-ROWS
INSERT INTO prod (name, id) VALUES (:NAME, :ID1) END-EXEC
* will process 25 rows
EXEC SQL FOR ARRAY_SIZE
INSERT INTO prod (name, id) VALUES (:NAME, :ID1) END-EXEC
```

expr

Specify the value to be inserted in the table. Array host variables, host variable literals, strings, and pointer variables can be specified. Structure type arrays and pointer variable arrays cannot be specified.

Do not use pointer variables and ARRAY_SIZE at the same time. The reason for this is that the number of elements in the area represented by the pointer variable cannot be determined.

query

A query (SELECT statement) that supplies the rows to be inserted. The number of rows returned by *query* must be 1. If two or more rows are returned, an error will occur. This cannot be used at the same time as ARRAY_SIZE.

outputHostVar and indicatorVal

These must be array host variables or pointer variables.

Error Messages

The messages below are output if an error occurs when the bulk INSERT is used.

Message

The value for the FOR clause must be a positive integer.

Cause

The value given for *numOfRows* is less than or equal to 0.

Solution

Specify a value that is more than or equal to 1 for *numOfRows*.

Message

Array host variable is needed when using FOR ARRAY_SIZE.

Cause

An array host variable is not specified in the VALUES clause.

Solution

Specify more than one array host variable in the VALUES clause.

Message

The SELECT..INTO query returned too many rows in row number %d.

Cause

The "SELECT ... INTO" query in the INSERT statement returned more than one row.

Solution

If *numOfRows* is more than one, the maximum number of rows that can be returned in the "SELECT ... INTO" query in the INSERT statement is one.

Limitations

The limitations when using bulk INSERT are given below.

- Array of structures should not be used as an input in the 'VALUES' clause.
- Array of pointers should not be used as an input in the 'VALUES' clause.
- ECOBPG supports the use of 'WITH' clause in single INSERT statements. 'WITH' clause cannot be used in bulk INSERT statements.
- If an error occurs, all bulk INSERT actions will be rolled back, therefore, no rows are inserted. However, if the RETURNING clause was used, and the error occurred while obtaining the rows after the insertion was successful, the insertion processing will not be rolled back.

Samples

Given below are some sample usages of the bulk INSERT functionality.

Basic Bulk INSERT

```
01 GROUP-ITEM.
05 IN-F1 PIC S9(9) OCCURS 4.
MOVE 1 TO IN-F1(1)
MOVE 2 TO IN-F1(2)
MOVE 3 TO IN-F1(3)
MOVE 4 TO IN-F1(4)
...
EXEC SQL FOR 3 INSERT INTO target (f1) VALUES (:IN-F1) END-EXEC
```

The number of rows to insert indicated by the FOR clause is 3, so the data in the first 3 elements of the host array variable are inserted into the table. The contents of the target table will be:

```
f1
----
1
2
3
(3 rows)
```

Also a host integer variable can be used to indicate the number of rows that will be inserted in FOR clause, which will produce the same result as above:

```
01 NUM PIC S9(9) COMP VALUE 3.
01 GROUP-ITEM.
05 IN-F1 PIC S9(9) OCCURS 4.
MOVE 1 TO IN-F1(1)
MOVE 2 TO IN-F1(2)
MOVE 3 TO IN-F1(3)
MOVE 4 TO IN-F1(4)
...
EXEC SQL FOR :NUM INSERT INTO target (f1) VALUES (:IN-F1) END-EXEC
```

Inserting constant values

Constant values can also be bulk INSERTed into the table as follows:

```
EXEC SQL FOR 3 INSERT INTO target (f1,f2) VALUES (DEFAULT,'hello') END-EXEC
```

Assuming the 'DEFAULT' value for the 'f1' column is '0', the contents of the target table will be:

```
f1 | f2
---+-------
0  | hello
0  | hello
0  | hello
(3 rows)
```

Using ARRAY_SIZE

'FOR ARRAY_SIZE' can be used to insert the entire contents of a host array variable, without explicitly specifying the size, into the table.

```
01 GROUP-ITEM.
05 IN-F1 PIC S9(9) OCCURS 4.
MOVE 1 TO IN-F1(1)
MOVE 2 TO IN-F1(2)
MOVE 3 TO IN-F1(3)
MOVE 4 TO IN-F1(4)
...
EXEC SQL FOR ARRAY_SIZE INSERT INTO target (f1) VALUES (:IN-F1) END-EXEC
```

Note

If there are multiple host array variables specified as input values, then the number of rows inserted is same as the smallest array size. The example given below demonstrates this usage.

```
01 GROUP-ITEM.
05 IN-F1 PIC S9(9) OCCURS 4.
05 IN-F3 PIC X(10) OCCURS 3.
MOVE 1 TO IN-F1(1)
MOVE 2 TO IN-F1(2)
MOVE 3 TO IN-F1(3)
MOVE 4 TO IN-F1(4)
MOVE "one" TO IN-F3(1)
MOVE "two" TO IN-F3(2)
MOVE "three" TO IN-F3(3)
...
EXEC SQL FOR ARRAY_SIZE INSERT INTO target (f1,f3) VALUES (:IN-F1,:IN-F3) END-EXEC
```

In the above example, the array sizes are 3 and 4. Given that the smallest array size is 3, only three rows are inserted into the table. The table contents are given below.

```
f1 | f3
---+-------
1  | one
2  | two
3  | three
(3 rows)
```

Using SELECT query

The result of a SELECT query can be used to insert values.

```
EXEC SQL FOR 4 INSERT INTO target(f1) SELECT age FROM source WHERE name LIKE 'foo' END-
EXEC
```

In the example above, assuming that the SELECT query returns one row, the same row will be inserted into the table four times.

Note

If "2" or more is specified for the FOR clause, the INSERT statement returns an error when two or more rows of query results are returned.

If "1" is specified for the FOR clause, all rows returned by the SELECT query will be inserted into the table.

```
EXEC SQL FOR 1 INSERT INTO target(f1) SELECT age FROM source END-EXEC
```

In the example above, "1" specified for the FOR clause indicates all returned rows.

Using RETURNING clause

Bulk INSERT supports the same RETURNING clause syntax as normal INSERT. An example is given below.

```
01 GROUP-ITEM.
05 IN-F1 PIC S9(9) OCCURS 4.
05 OUT-F1 PIC S9(9) OCCURS 4.
MOVE 1 TO IN-F1(1)
MOVE 2 TO IN-F1(2)
MOVE 3 TO IN-F1(3)
MOVE 4 TO IN-F1(4)
...
EXEC SQL FOR 3 INSERT INTO target (f1) VALUES (:IN-F1) RETURNING f1 INTO :OUT-F1 END-EXEC
```

After the execution of the above INSERT statement, the 'out_f1' array will have 3 elements with the values of '1', '2' and '3'.

7.4.4 DECLARE STATEMENT

Refer to "DECLARE STATEMENT" in the PostgreSQL Documentation.

7.4.5 Creating Applications while in Database Multiplexing Mode

This section explains points to consider when creating applications while in database multiplexing mode.

See

- Refer to the Cluster Operation Guide (Database Multiplexing) for information on database multiplexing mode.

- Refer to "Application Development" in the Cluster Operation Guide (PRIMECLUSTER) for points to consider when creating applications using the failover feature integrated with the cluster software.

7.4.5.1 Errors when an Application Connection Switch Occurs and Corresponding Actions

If an application connection switch occurs while in database multiplexing mode, explicitly close the connection and then reestablish the connection or reexecute the application.

The table below shows errors that may occur during a switch, and the corresponding action to take.

State		Error information (*1)	Action
Server failure or Fujitsu Enterprise Postgres system failure	Failure occurs during access	57P01 57P02 YE000 26000 40001	After the switch is complete, reestablish the connection, or reexecute the application.
	Accessed during system failure	08001	
Switch to the standby server	Switched during access	57P01 57P02 YE000 26000 40001	
	Accessed during switch	08001	

*1: Return value of SQLSTATE.

Chapter 8 SQL References

This chapter explains the SQL statement features expanded by Fujitsu Enterprise Postgres.

8.1 Expanded Trigger Definition Feature

This section explains the expanded trigger definition feature.

8.1.1 CREATE TRIGGER

In addition to features of PostgreSQL, triggers can be created with DO option.

Synopsis

```
CREATE [ OR REPLACE ] [ CONSTRAINT ] TRIGGER name { BEFORE | AFTER | INSTEAD OF } { event
[ OR ... ] }
    ON table_name
    [ FROM referenced_table_name ]
    [ NOT DEFERRABLE | [ DEFERRABLE ] [ INITIALLY IMMEDIATE | INITIALLY DEFERRED ] ]
    [ REFERENCING { { OLD | NEW } TABLE [ AS ] transition_relation_name } [ ... ] ]
    [ FOR [ EACH ] { ROW | STATEMENT } ]
    [ WHEN ( condition ) ]
    { EXECUTE { FUNCTION | PROCEDURE } function_name ( arguments )
        | DO [ LANGUAGE lang_name ] code }
```

Description

Refer to the PostgreSQL Documentation for information about CREATE TRIGGER. This section describes DO option.

A trigger which is created with DO option will be associated with the specified table or view and will execute the specified code by the specified procedural language of DO (unnamed code block) when certain events occur.

Parameters

lang_name

The name of the language that the function is implemented in.

plpgsql is supported in CREATE TRIGGER.

code

When the certain events occur, it executes the code in a specified procedural language. The unnamed code block does not require a prior definition like a function. Syntax is same as procedural language.

Note

- A trigger defined with DO option cannot be replaced by a trigger defined with EXECUTE PROCEDURE option.
- A trigger defined with EXECUTE PROCEDURE option cannot be replaced by a trigger defined with DO option.

Examples

It executes the code block that is specified by DO before the table is updated.
(Example that LANGUAGE is plpgsql)

```
CREATE TRIGGER check_update
    BEFORE UPDATE ON accounts
```

```
    FOR EACH ROW
    DO $$BEGIN RETURN NEW; END;$$ ;
```

Information

When a trigger created with DO option, a new function is created internally. The name of function is "schema name"."on table name"_"trigger name"_TRIGPROC(serial number).

8.1.2 How to Define Triggers in pgAdmin

The expanded features of the trigger definition can also be used in pgAdmin.

See

Refer to "pgAdmin Help" for information on how to define triggers using pgAdmin.

Chapter 9 Compatibility with Oracle Databases

This chapter describes the environment settings and functionality offered for features that are compatible with Oracle databases.

9.1 Overview

Features compatible with Oracle databases are provided. These features enable you to easily migrate to Fujitsu Enterprise Postgres and reduce the costs of reconfiguring applications.

The table below lists features compatible with Oracle databases.

Table 9.1 Features compatible with Oracle databases

Category		Feature	
		Item	Overview
SQL	Queries	Outer join operator (+)	Operator for outer joins
		DUAL table	Table provided by the system
	Functions	DECODE	Compares values, and if they match, returns a corresponding value
		SUBSTR	Extracts part of a string using characters to specify position and length
		NVL	Returns a substitute value when a value is NULL
Package		DBMS_ALERT	Sends alerts
		DBMS_ASSERT	Perform assertions on input values
		DBMS_OUTPUT	Sends messages to clients
		DBMS_PIPE	Execution of inter-session communication
		DBMS_RANDOM	Random number generation
		DBMS_UTILITY	Addition of various functions
		UTL_FILE	Enables text file operations
		DBMS_SQL	Enables dynamic SQL execution

See

In addition to the above, refer to the file below for information on the features compatible with Oracle databases.

L - Linux:
fujitsuEnterprisePostgresInstallDir/share/doc/extension/README.asciidoc

W - Windows(R):
fujitsuEnterprisePostgresInstallDir\doc\extension\README.asciidoc

9.2 Precautions when Using the Features Compatible with Oracle Databases

This section provides notes on using the features compatible with oracle databases.

9.2.1 Notes on search_path

Objects defined with the features compatible with oracle databases are defined in the oracle schema. Therefore, when using this function, it is necessary to add "oracle" to the "search_path" parameter in postgresql.conf.

```
search_path = '"$user", public, oracle'
```

Information

- The search_path parameter specifies the order in which schemas are searched.
- Refer to "Statement Behavior" in "Client Connection Defaults" in "Server Administration" in the PostgreSQL Documentation for information on search_path.

9.2.2 Notes on SUBSTR

SUBSTR is implemented in Fujitsu Enterprise Postgres and Oracle databases using different external specifications.

For this reason, when using SUBSTR, define which specification is to take precedence. In the default configuration of Fujitsu Enterprise Postgres, the specifications of Fujitsu Enterprise Postgres take precedence.

When using the SUBSTR function compatible with Oracle databases, set "oracle" and "pg_catalog" in the "search_path" parameter of postgresql.conf. You must specify "oracle" before "pg_catalog" when doing this.

```
search_path = '"$user", public, oracle, pg_catalog'
```

9.2.3 Notes when Integrating with the Interface for Application Development

The SQL noted in "Table 9.1 Features compatible with Oracle databases" can be used in the interface for application development.

Note that both "public" and the schema name in the SQL statement must be specified as the SearchPath parameter before "oracle" and "pg_catalog" when using the Oracle database-compatible feature SUBSTR.

9.3 Queries

The following queries are supported:

- Outer Join Operator (+)
- DUAL Table

9.3.1 Outer Join Operator (+)

In the WHERE clause conditional expression, by adding the plus sign (+), which is the outer join operator, to the column of the table you want to add as a table join, it is possible to achieve an outer join that is the same as a joined table (OUTER JOIN).

Syntax

SELECT statement

```
SELECT … [WHERE [NOT] joinCond ...] …
SELECT … [WHERE srchCond ]... ] …
```

Join condition

```
{ colSpec(+) = colSpec | colSpec = colSpec(+) }
```

Note

Here we are dealing only with the WHERE clause of the SELECT statement. Refer to "SQL Commands" in "Reference" in the PostgreSQL Documentation for information on the overall syntax of the SELECT statement.

General rules

WHERE clause

- The WHERE clause specifies search condition or join conditions for the tables that are derived.
- Search conditions are any expressions that return BOOLEAN types as the results of evaluation. Any rows that do not meet these conditions are excluded from the output. When the values of the actual rows are assigned to variables and if the expression returns TRUE, those rows are considered to have met the conditions.
- Join conditions are comparison conditions that specify outer join operators. Join conditions in a WHERE clause return a table that includes all the rows that meet the join conditions, including rows that do not meet all the join conditions.
- Join conditions take precedence over search conditions. For this reason, all rows returned by the join conditions are subject to the search conditions.
- The following rules and restrictions apply to queries that use outer join operators. It is therefore recommended to use FROM clause joined tables (OUTER JOIN) rather than outer join operators:
  - Outer join operators can only be specified in the WHERE clause.
  - Outer join operators can only be specified for base tables or views.
  - To perform outer joins using multiple join conditions, it is necessary to specify outer join operators for all join conditions.
  - When combining join conditions with constants, specify outer join operators in the corresponding column specification. When not specified, they will be treated as search conditions.
  - The results column of the outer join of table t1 is not returned if table t1 is joined with table t2 by specifying an outer join operator in the column of t1, then table t1 is joined with table t3 by using search conditions.
  - It is not possible to specify columns in the same table as the left/right column specification of a join condition.
  - It is not possible to specify an expression other than a column specification for outer join operators, but they may be specified for the columns that compose the expression.

    There are the following limitations on the functionality of outer join operators when compared with joined tables (OUTER JOIN). To use functionality that is not available with outer join operators, use joined tables (OUTER JOIN).

Table 9.2 Range of functionality with outer join operators

Functionality available with joined tables (OUTER JOIN)	Outer join operator
Outer joins of two tables	Y
Outer joins of three or more tables	Y (*1)
Used together with joined tables within the same query	N
Use of the OR logical operator to a join condition	N
Use of an IN predicate to a join condition	N
Use of subqueries to a join condition	N

Y: Available
N: Not available

*1: The outer joins by outer join operators can return outer join results only for one other table. For this reason, to combine outer joins of table t1 and table t2 or table t2 and table t3, it is not possible to specify outer join operators simultaneously for table t2.

Example

Table configuration

t1

col1	col2	col3
1001	AAAAA	1000
1002	BBBBB	2000
1003	CCCCC	3000

t2

col1	col2
1001	aaaaa
1002	bbbbb
1004	ddddd

Example 1: Return all rows in table t2, including those that do not exist in table t1.

```
SELECT *
  FROM t1, t2
  WHERE t1.col1(+) = t2.col1;
 col1 |    col2    | col3 | col1 |    col2
------+------------+------+------+------------
 1001 | AAAAA      | 1000 | 1001 | aaaaa
 1002 | BBBBB      | 2000 | 1002 | bbbbb
      |            |      | 1004 | ddddd
(3 rows)
```

This is the same syntax as the joined table (OUTER JOIN) of the FROM clause shown next.

```
SELECT *
  FROM t1 RIGHT OUTER JOIN t2
       ON t1.col1 = t2.col1;
```

Example 2: In the following example, the results are filtered to records above 2000 in t1.col3 by search conditions, and the records are those in table t2 that include ones that do not exist in table t1. After filtering with the join conditions, there is further filtering with the search conditions, so there will only be one record returned.

```
SELECT *
  FROM t1, t2
  WHERE t1.col1(+) = t2.col1
    AND t1.col3 >= 2000;
 col1 |    col2    | col3 | col1 |    col2
------+------------+------+------+------------
 1002 | BBBBB      | 2000 | 1002 | bbbbb
(1 row)
```

This is the same syntax as the joined table (OUTER JOIN) of the FROM clause shown next.

```
SELECT *
  FROM t1 RIGHT OUTER JOIN t2
       ON t1.col1 = t2.col1
  WHERE t1.col3 >= 2000;
```

9.3.2 DUAL Table

DUAL table is a virtual table provided by the system. Use when executing SQL where access to a base table is not required, such as when performing tests to get result expressions such as functions and operators.

Example

In the following example, the current system date is returned.

```
SELECT  CURRENT_DATE  "date" FROM DUAL;
    date
------------
 2013-05-14
(1 row)
```

9.4 SQL Function Reference

The following SQL functions are supported:

- DECODE
- SUBSTR
- NVL

9.4.1 DECODE

Description

Compares values and if they match, returns a corresponding value.

Syntax

```
DECODE(expr, srch, result [, srch, result ]... [, default ])
```

General rules

- DECODE compares values of the value expression to be converted and the search values one by one. If the values match, a corresponding result value is returned. If no values match, the default value is returned if it has been specified. A NULL value is returned if a default value has not been specified.
- If the same search value is specified more than once, then the result value returned is the one listed for the first occurrence of the search value.
- The following data types can be used in result values and in the default value:
  - CHAR
  - VARCHAR
  - NCHAR
  - NCHAR VARYING
  - TEXT
  - INTEGER
  - BIGINT

  - NUMERIC
  - DATE
  - TIME WITHOUT TIME ZONE
  - TIMESTAMP WITHOUT TIME ZONE
  - TIMESTAMP WITH TIME ZONE
- The same data type must be specified for the values to be converted and the search values. However, note that different data types may also be specified if a literal is specified in the search value, and the value expression to be converted contains data types that can be converted. When specifying literals, refer to "Table A.1 Data type combinations that contain literals and can be converted implicitly" in "A.3 Implicit Data Type Conversions" for information on the data types that can be specified.
- If the result values and default value are all literals, the data types for these values will be as shown below:
  - If all values are string literals, all will become character types.
  - If there is one or more numeric literal, all will become numeric types.
  - If there is one or more literal cast to the datetime/time types, all will become datetime/time types.
- If the result values and default value contain a mixture of literals and non-literals, the literals will be converted to the data types of the non-literals. When specifying literals, refer to "Table A.1 Data type combinations that contain literals and can be converted implicitly" in "A.3 Implicit Data Type Conversions" for information on the data types that can be converted.
- The same data type must be specified for all result values and for the default value. However, different data types can be specified if the data type of any of the result values or default value can be converted - these data types are listed below:

Table 9.3 Data type combinations that can be converted by DECODE (summary)

		Other result values or default value		
		Numeric type	Character type	Date/time type
Result value (any)	Numeric type	Y	N	N
	Character type	N	Y	N
	Date/time type	N	N	S (*1)

Y: Can be converted
S: Some data types can be converted
N: Cannot be converted

*1: The data types that can be converted for date/time types are listed below:

Table 9.4 Result value and default value date/time data types that can be converted by DECODE

		Other result values or default value			
		DATE	TIME WITHOUT TIME ZONE	TIMESTAMP WITHOUT TIME ZONE	TIMESTAMP WITH TIME ZONE
Result value (any)	DATE	Y	N	Y	Y
	TIME WITHOUT TIME ZONE	N	Y	N	N
	TIMESTAMP WITHOUT TIME ZONE	Y	N	Y	Y
	TIMESTAMP WITH TIME ZONE	Y	N	Y	Y

Y: Can be converted
N: Cannot be converted

- The data type of the return value will be the data type within the result or default value that is longest and has the highest precision.

Example

In the following example, the value of col3 in table t1 is compared and converted to a different value. If the col3 value matches search value 1, the result value returned is "one". If the col3 value does not match any of search values 1, 2, or 3, the default value "other number" is returned.

```
SELECT col1, DECODE(col3, 1000, 'one',
                                2000, 'two',
                                3000, 'three',
                                'other number') "num-word"
       FROM t1;
col1  | num-word
------+----------
 1001 | one
 1002 | two
 1003 | three
(3 rows)
```

9.4.2 SUBSTR

Description

Extracts part of a string using characters to specify position and length.

Syntax

```
SUBSTR(str, startPos [, len ])
```

General rules

- SUBSTR extracts and returns a substring of string *str*, beginning at position *startPos*, for number of characters *len*.
- When *startPos* is positive, it will be the number of characters from the beginning of the string.
- When *startPos* is 0, it will be treated as 1.
- When *startPos* is negative, it will be the number of characters from the end of the string.
- When *len* is not specified, all characters to the end of the string are returned. NULL is returned when *len* is less than 1.
- For *startPos* and *len*, specify a SMALLINT or INTEGER type. When specifying literals, refer to "Table A.1 Data type combinations that contain literals and can be converted implicitly" in "A.3 Implicit Data Type Conversions" for information on the data types that can be specified.
- The data type of the return value is TEXT.

Note

- There are two types of SUBSTR. One that behaves as described above, and one that behaves the same as SUBSTRING. The search_path parameter must be modified for it to behave the same as the specification described above.
- It is recommended to set search_path in postgresql.conf. In this case, it will be effective for each instance. Refer to "9.2.2 Notes on SUBSTR" for information on how to configure postgresql.conf.
- The configuration of search_path can be done at the user level or at the database level. Setting examples are shown below.
  - Example of setting at the user level

    This can be set by executing an SQL command. In this example, user1 is used as the username.

```
ALTER USER user1 SET search_path = "$user",public,oracle,pg_catalog;
```

- Example of setting at the database level

  This can be set by executing an SQL command. In this example, db1 will be used as the database name.

```
ALTER DATABASE db1 SET search_path = "$user",public,oracle,pg_catalog;
```

  You must specify "oracle" before "pg_catalog".

- If the change has not been implemented, SUBSTR is the same as SUBSTRING.

See

Refer to "SQL Commands" in "Reference" in the PostgreSQL Documentation for information on ALTER USER and ALTER DATABASE.

Information

The general rules for SUBSTRING are as follows:

- The start position will be from the beginning of the string, whether positive, 0, or negative.
- When *len* is not specified, all characters to the end of the string are returned.
- An empty string is returned if no string is extracted or *len* is less than 1.

See

Refer to "String Functions and Operators" under "The SQL Language" in the PostgreSQL Documentation for information on SUBSTRING.

Example

In the following example, part of the string "ABCDEFG" is extracted:

```
SELECT SUBSTR('ABCDEFG',3,4) "Substring" FROM DUAL;

 Substring
-----------
 CDEF
(1 row)

SELECT SUBSTR('ABCDEFG',-5,4) "Substring" FROM DUAL;

 Substring
-----------
(1 row)
```

9.4.3 NVL

Description

Returns a substitute value when a value is NULL.

Syntax

```
NVL(expr1, expr2)
```

General rules

- NVL returns a substitute value when the specified value is NULL. When *expr1* is NULL, *expr2* is returned. When *expr1* is not NULL, *expr1* is returned.
- Specify the same data types for *expr1* and *expr2*. However, if a constant is specified in *expr2*, and the data type can also be converted by *expr1*, different data types can be specified. When this happens, the conversion by *expr2* is done to suit the data type in *expr1*, so the value of *expr2* returned when *expr1* is a NULL value will be the value converted in the data type of *expr1*.
- When specifying literals, refer to "Table A.1 Data type combinations that contain literals and can be converted implicitly" in "A.3 Implicit Data Type Conversions" for information on the data types that can be converted.

Example

In the following example, "IS NULL" is returned if the value of col1 in table t1 is a NULL value.

```
SELECT col2, NVL(col1,'IS NULL') "nvl" FROM t1;
 col2 |   nvl
------+---------
 aaa  | IS NULL
(1 row)
```

9.5 Package Reference

A "package" is a group of features, brought together by schemas, that have a single functionality, and are used by calling from PL/pgSQL.

The following packages are supported:

- DBMS_ALERT
- DBMS_ASSERTION
- DBMS_OUTPUT
- DBMS_PIPE
- DBMS_RANDOM
- DBMS_UTILITUY
- UTL_FILE
- DBMS_SQL

To call each feature from PL/pgSQL, use the PERFORM or SELECT statement and qualify the feature name with the package name. For more information on the calling format, refer to the feature-specific description for each package.

In the following, explain DBMS_SQL among the supported packages. For other packages, refer to the README stored in the installation location.

See

For packages other than DBMS_SQL, refer to the following file.

L

- Linux:
  fujitsuEnterprisePostgresInstallDir/share/doc/extension/README.asciidoc

- Windows(R):
  fujitsuEnterprisePostgresInstallDir\doc\extension\README.asciidoc

9.5.1 DBMS_SQL

Overview

Dynamic SQL can be executed from PL/pgSQL.

Features

Feature	Description
BIND_VARIABLE	Sets values in the host variable within the SQL statement.
CLOSE_CURSOR	Closes the cursor.
COLUMN_VALUE	Retrieves the value of the column in the select list extracted with FETCH_ROWS.
DEFINE_COLUMN	Defines the column from which values are extracted and the storage destination.
EXECUTE	Executes SQL statements.
FETCH_ROWS	Positions the specified cursor at the next row and extracts values from the row.
OPEN_CURSOR	Opens a new cursor.
PARSE	Parses SQL statements.

Note

- In DBMS_SQL, the data types supported in dynamic SQL are limited, and therefore the user must consider this. The supported data types are:
  - INTEGER
  - DECIMAL
  - NUMERIC
  - REAL
  - DOUBLE PRECISION
  - CHAR(*1)
  - VARCHAR(*1)
  - NCHAR(*1)
  - NCHAR VARYING(*1)
  - TEXT
  - DATE
  - TIMESTAMP WITHOUT TIME ZONE
  - TIMESTAMP WITH TIME ZONE
  - INTERVAL(*2)
  - SMALLINT

- BIGINT

*1:
The host variables with CHAR, VARCHAR, NCHAR, and NCHAR VARYING data types are treated as TEXT, to match the string function arguments and return values. Refer to "String Functions and Operators" in "Functions and Operators" in "The SQL Language" in the PostgreSQL Documentation for information on string functions.
When specifying the arguments of the features compatible with Oracle databases NVL and/or DECODE, use CAST to convert the data types of the host variables to ensure that data types between arguments are the same.

*2:
When using COLUMN_VALUE to obtain an INTERVAL type value specified in the select list, use an INTERVAL type variable with a wide range such as when no interval qualifier is specified, or with a range that matches that of the variable in the select list. If an interval qualifier variable with a narrow range is specified, then the value within the interval qualifier range will be obtained, but an error that the values outside the range have been truncated will not occur.

Example

This example illustrates where a value expression that returns an INTERVAL value is set in the select list and the result is received with COLUMN_VALUE. Note that the SQL statement operation result returns a value within the INTERVAL DAY TO SECOND range.

[Bad example]

Values of MINUTE, and those after MINUTE, are truncated, because the variable(v_interval) is INTERVAL DAY TO HOUR.

```
    v_interval     INTERVAL DAY TO HOUR;
...
    PERFORM DBMS_SQL.PARSE(cursor, 'SELECT CURRENT_TIMESTAMP - ''2010-01-01'' FROM
DUAL', 1);
...
    SELECT value INTO v_interval FROM DBMS_SQL.COLUMN_VALUE(cursor, 1, v_interval);
    result:1324 days 09:00:00
```

[Good example]

By ensuring that the variable(v_interval) is INTERVAL, the values are received correctly.

```
    v_interval     INTERVAL;
...
    PERFORM DBMS_SQL.PARSE(cursor, 'SELECT CURRENT_TIMESTAMP - ''2010-01-01'' FROM
DUAL', 1);
...
    SELECT value INTO v_interval FROM DBMS_SQL.COLUMN_VALUE(cursor, 1, v_interval);
    result:1324 days 09:04:37.530623
```

Syntax

```
{ BIND_VARIABLE(cursor, varName, val [, len ])
| CLOSE_CURSOR(cursor)
| COLUMN_VALUE(cursor, colPos, varName)
| DEFINE_COLUMN(cursor, colPos, varName [, len ])
| EXECUTE(cursor)
| FETCH_ROWS(cursor)
| OPEN_CURSOR([parm1 ])
| PARSE(cursor, sqlStmt, parm1 [, parm2, parm3, parm4 ])
}
```

9.5.1.1 Description

This section explains each feature of DBMS_SQL.

BIND_VARIABLE

- BIND_VARIABLE sets values in the host variable within the SQL statement.
- Specify the cursor number to be processed.
- Specify the name of the host variable within the SQL statement using a string for the host variable name.
- Specify the value set in the host variable. The data type of the host variable is the same as that of the value expression - it is implicitly converted in accordance with its position within the SQL statement. Refer to "A.3 Implicit Data Type Conversions" for information on implicit conversions.
- If the value is a character type, the string length is the number of characters. If the string length is not specified, the size is the total length of the string.
- It is necessary to place a colon at the beginning of the host variable in SQL statements to identify the host variable. The colon does not have to be added to the host variable names specified at BIND_VARIABLE. The following shows examples of host variable names specified with SQL statements and host variable names specified with BIND_VARIABLE:

```
PERFORM DBMS_SQL.PARSE(cursor, 'SELECT emp_name FROM emp WHERE sal > :x', 1);
```

  In this example, BIND_VARIABLE will be as follows:

```
PERFORM DBMS_SQL.BIND_VARIABLE(cursor, ':x', 3500);
```

  Or,

```
PERFORM DBMS_SQL.BIND_VARIABLE(cursor, 'x', 3500);
```

- The length of the host variable name can be up to 30 bytes (excluding colons).
- If the data type of the set value is string, specify the effective size of the column value as the fourth argument.

Example

If the data type of the value to be set is not a string:

```
PERFORM DBMS_SQL.BIND_VARIABLE(cursor, ':NO', 1);
```

If the data type of the value to be set is a string:

```
PERFORM DBMS_SQL.BIND_VARIABLE(cursor, ':NAME', h_memid, 5);
```

CLOSE_CURSOR

- CLOSE_CURSOR closes the cursor.
- Specify the cursor number to be processed.
- The value returned is a NULL value.

Example

```
cursor := DBMS_SQL.CLOSE_CURSOR(cursor);
```

COLUMN_VALUE

- COLUMN_VALUE retrieves the value of the column in the select list extracted with FETCH_ROWS.

- Specify the cursor number to be processed.
- Specify the position of the column of the select list in the SELECT statement. The position of the first column is 1.
- Specify the destination variable name.
- Use a SELECT statement to obtain the values of the value, column_error, and actual_length columns.
- The value column returns the value of the column specified at the column position. The data type of the variable name must match that of the column. If the data type of the column in the SELECT statement specified in PARSE is not compatible with DBMS_SQL, use CAST to convert to a compatible data type.
- The data type of the column_error column is NUMERIC. If the column value could not be set correctly in the value column, a value other than 0 will be returned:
  22001: The extracted string has been truncated
  22002: The extracted value contains a NULL value
- The data type of the actual_length column is INTEGER. If the extracted value is a character type, the number of characters will be returned (if the value was truncated, the number of characters prior to the truncation will be returned), otherwise, the number of bytes will be returned.

Example

When retrieving the value of the column, the error code, and the actual length of the column value:

```
SELECT value, column_error, actual_length INTO v_memid, v_col_err, v_act_len FROM
DBMS_SQL.COLUMN_VALUE(cursor, 1, v_memid);
```

When retrieving just the value of the column:

```
SELECT value INTO v_memid FROM DBMS_SQL.COLUMN_VALUE(cursor, 1, v_memid);
```

DEFINE_COLUMN

- DEFINE_COLUMN defines the column from which values are extracted and the storage destination.
- Specify the cursor number to be processed.
- Specify the position of the column in the select list in the SELECT statement. The position of the first column is 1.
- Specify the destination variable name. The data type should be match with the data type of the column from which the value is to be extracted. If the data type of the column in the SELECT statement specified in PARSE is not compatible with DBMS_SQL, use CAST to convert to a compatible data type.
- Specify the maximum number of characters of character type column values.
- If the data type of the column value is string, specify the effective size of the column value as the fourth argument.

Example

When the data type of the column value is not a string:

```
PERFORM DBMS_SQL.DEFINE_COLUMN(cursor, 1, v_memid);
```

When the data type of the column value is a string:

```
PERFORM DBMS_SQL.DEFINE_COLUMN(cursor, 1, v_memid, 10);
```

EXECUTE

- EXECUTE executes SQL statements.

- Specify the cursor number to be processed.
- The return value is an INTEGER type, is valid only with INSERT statement, UPDATE statement, and DELETE statement, and is the number of rows processed. Anything else is invalid.

Example

```
ret := DBMS_SQL.EXECUTE(cursor);
```

FETCH_ROWS

- FETCH_ROWS positions at the next row and extracts values from the row.
- Specify the cursor number to be processed.
- The return value is an INTEGER type and is the number of rows extracted. 0 is returned if all are extracted.
- The extracted information is retrieved with COLUMN_VALUE.

Example

```
LOOP
    IF DBMS_SQL.FETCH_ROWS(cursor) = 0 THEN
        EXIT;
    END IF;

...

END LOOP;
```

OPEN_CURSOR

- OPEN_CURSOR opens a new cursor.
- The parameter is used for compatibility with Oracle databases only, and is ignored by Fujitsu Enterprise Postgres. An INTEGER type can be specified, but it will be ignored. If migrating from an Oracle database, specify 1.
- Close unnecessary cursors by executing CLOSE_CURSOR.
- The return value is an INTEGER type and is the cursor number.

Example

```
cursor := DBMS_SQL.OPEN_CURSOR();
```

PARSE

- PARSE analyzes dynamic SQL statements.
- Specify the cursor number to be processed.
- Specify the SQL statement to be parsed.
- Parameters 1, 2, 3, and 4 are used for compatibility with Oracle databases only, and are ignored by Fujitsu Enterprise Postgres. If you are specifying values anyway, specify the following:
  - Parameter 1 is an INTEGER type. Specify 1.

- Parameters 2 and 3 are TEXT types. Specify NULL.
- Parameter 4 is a BOOLEAN type. Specify TRUE.

If migrating from an Oracle database, the specified values for parameters 2, 3, and 4 do not need to be changed.

- Add a colon to the beginning of host variables in SQL statements.
- The DDL statement is executed when PARSE is issued. EXECUTE is not required for the DDL statement.
- If PARSE is called again for opened cursors, the content in the data regions within the cursors is reset, and the SQL statement is parsed anew.

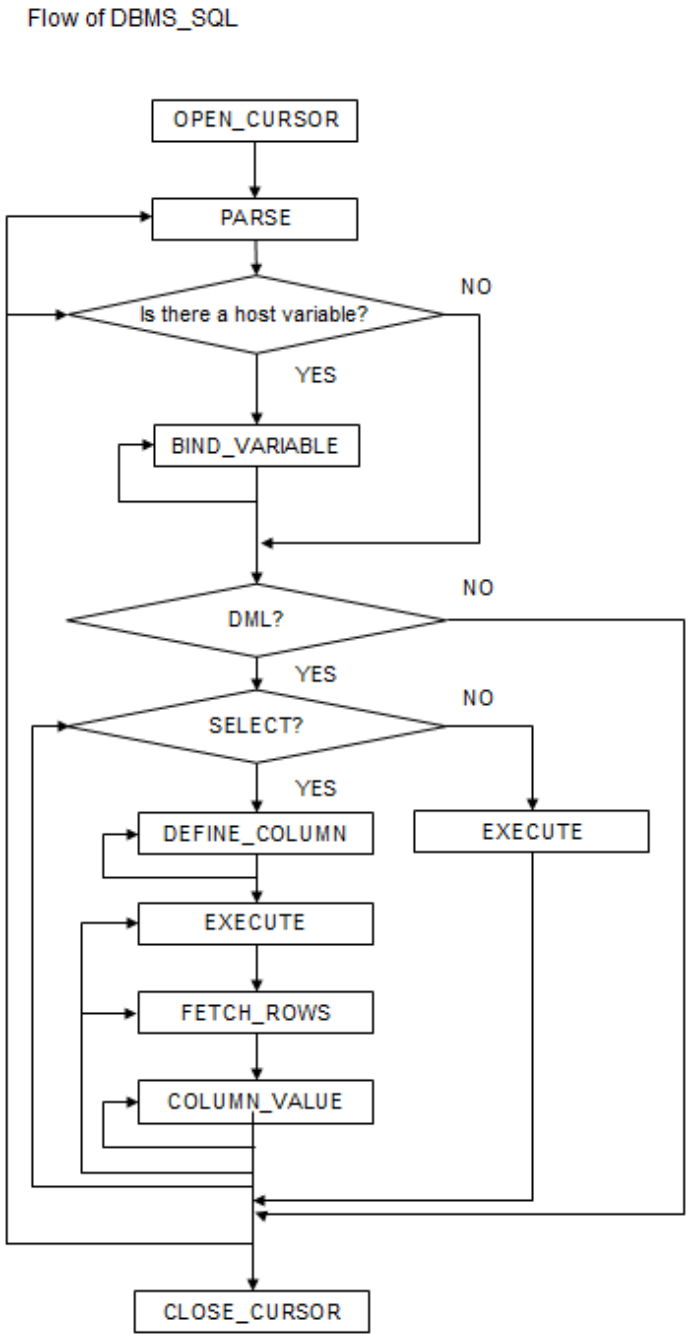
Example

```
PERFORM DBMS_SQL.PARSE(cursor, 'SELECT memid, memnm FROM member WHERE memid = :NO', 1);
```

9.5.1.2 Example

This section explains the flow of DBMS_SQL and provides an example.

Flow of DBMS_SQL



Example

```
CREATE FUNCTION smp_00()
RETURNS INTEGER
AS $$
DECLARE
    str_sql     VARCHAR(255);
    cursor      INTEGER;
    h_smpid     INTEGER;
    v_smpid     INTEGER;
    v_smpnm     VARCHAR(20);
    v_smpage    INTEGER;
    errcd       INTEGER;
```

```
    length      INTEGER;
    ret         INTEGER;
BEGIN
   str_sql    := 'SELECT smpid, smpnm, smpage FROM smp_tbl WHERE smpid < :H_SMPID ORDER BY
smpid';
    h_smpid     := 3;
    v_smpid     := 0;
    v_smpnm     := '';
    v_smpage    := 0;

    cursor := DBMS_SQL.OPEN_CURSOR();

    PERFORM DBMS_SQL.PARSE(cursor, str_sql, 1);

    PERFORM DBMS_SQL.BIND_VARIABLE(cursor, ':H_SMPID', h_smpid);

    PERFORM DBMS_SQL.DEFINE_COLUMN(cursor, 1, v_smpid);
    PERFORM DBMS_SQL.DEFINE_COLUMN(cursor, 2, v_smpnm, 10);
    PERFORM DBMS_SQL.DEFINE_COLUMN(cursor, 3, v_smpage);

    ret := DBMS_SQL.EXECUTE(cursor);
    loop
        if DBMS_SQL.FETCH_ROWS(cursor) = 0 then
            EXIT;
        end if;

        SELECT value,column_error,actual_length INTO v_smpid,errcd,length FROM
DBMS_SQL.COLUMN_VALUE(cursor, 1, v_smpid);
        RAISE NOTICE '------------------------------------------------------';
        RAISE NOTICE '------------------------------------------------------';
        RAISE NOTICE 'smpid       = %', v_smpid;
        RAISE NOTICE 'errcd       = %', errcd;
        RAISE NOTICE 'length      = %', length;

        SELECT value,column_error,actual_length INTO v_smpnm,errcd,length FROM
DBMS_SQL.COLUMN_VALUE(cursor, 2, v_smpnm);
        RAISE NOTICE '------------------------------------------------------';
        RAISE NOTICE 'smpnm       = %', v_smpnm;
        RAISE NOTICE 'errcd       = %', errcd;
        RAISE NOTICE 'length      = %', length;

        select value,column_error,actual_length INTO v_smpage,errcd,length FROM
DBMS_SQL.COLUMN_VALUE(cursor, 3, v_smpage);
        RAISE NOTICE '------------------------------------------------------';
        RAISE NOTICE 'smpage      = %', v_smpage;
        RAISE NOTICE 'errcd       = %', errcd;
        RAISE NOTICE 'length      = %', length;
        RAISE NOTICE '';
    end loop;

    cursor := DBMS_SQL.CLOSE_CURSOR(cursor);
    RETURN 0;
END;
$$ LANGUAGE plpgsql;
```

Chapter 10 Application Connection Switch Feature

The application connection switch feature enables automatic connection to the target server when there are multiple servers with redundant configurations.

When using this feature, specify the primary server and secondary server as the connected servers in the application connection information. A standby server can optionally be prioritized over the primary server as the target server.

If an application connection switch occurs, explicitly close the connection and then reestablish the connection or reexecute the application. Refer to "Errors when an Application Connection Switch Occurs and Corresponding Actions" of the relevant client interface for information on how to confirm the switch.

10.1 Connection Information for the Application Connection Switch Feature

To use the application connection switch feature, set the information shown below when connecting the database.

IP address or host name

Specify the IP address or host name that will be used to configure the database multiplexing system.

Port number

A port number used by each database server to listen for connections from applications.
In each client interface, multiple port numbers can be specified, however in the format shown below, for example:

host1,host2:port2

JDBC and .NET

If only one port number is specified, it will be assumed that host1: 27500 (the default value) and host2:port2 were specified.
Omit all port numbers, or specify only one per server.

Others

If only one port number is specified, it will be assumed that the same port is used for all the hosts.

Target server

From the specified connection destination server information, specify the selection sequence of the servers to which the application will connect. The values specified for the target server have the meanings shown below. If a value is omitted, "any" will be assumed.

Primary server

The primary server is selected as the connection target from the specified "IP addresses or host names". Specify this to perform tasks that can be performed only on the primary server, such as applications in line with updates, or management tasks such as REINDEX and VACUUM.

Standby server

The standby server is selected as the connection target from the specified "IP addresses or host names". On standby server, the update will always fail. If the target server is not standby, the JDBC driver will throw an error stating that it is unable to find a server with the specified targetServerType.

Priority given to a primary server

The primary server is selected preferentially as the connection target from the specified "IP addresses or host names". If there is no primary server, the application will connect to the standby server.

Priority given to a standby server

The standby server is selected preferentially as the connection target from the specified "IP addresses or host names". If there is no standby server, the application will connect to the primary server.

Any

This method is not recommended in database multiplexing systems. This is because, although the connection destination server is selected in the specified sequence from the specified "IP addresses or host names", if the server that was successfully connected to first is the standby server, the write operations will always fail.

The table below shows the server selection order values to set for each driver:

Server selection order	JDBC drivers	.NET Data Provider	Other drivers
Primary server	"primary"(*1)	"read-write"(*1) "primary"(*2)	"read-write"(*1) "primary"(*2)
Standby server	"secondary"(*2)	"standby" "read-only"(*2)	"standby" "read-only"(*2)
Priority given to a primary server	"preferPrimary"(*3)	"prefer-primary"	-
Priority given to a standby server	"preferSecondary"(*2)	"prefer-standby"	"prefer-standby"(*4)
Any	"any"	"any"	"any"

*1: The primary server whose default transaction mode is read-only is not selected.

*2: The primary server whose default transaction mode is read-only is also selected.

*3: Prefer primary servers whose default transaction mode is read-write.

*4: While it is possible to specify "prefer-read" instead of "prefer-standby" for compatibility with Fujitsu Enterprise Postgres 13 and earlier, this is not recommended.

SSL server certificate Common Name (CN)

To perform SSL authentication by creating the same server certificate for each server in a multiplexing system, specify the SSL server certificate Common Name (CN) in this parameter. Accordingly, SSL authentication using the CN can be performed without having to consider the names of the multiple servers contained in the multiplexing system.

10.2 Using the Application Connection Switch Feature

This section explains how to set the connection destination server using the application connection switch feature.

Of the parameters used as connection information for each client interface, only the parameters specific to the application connection switch feature are explained here. Refer to "Setup" and "Connecting to the Database" for information on the other parameters of each client interface.

10.2.1 Using the JDBC Driver

Set the following information in the connection string of the DriverManager class, or in the data source.

Table 10.1 Information to be set

Argument	Explanation
host1 host2	Specify the IP address or host name. The IP address or host name can be omitted. If omitted, the default is localhost.
port1 port2	Specify the port number for the connection. The port number can be omitted. If omitted, the default is 27500.
database_name	Specify the database name.
targetServerType	Specify the selection sequence of the servers to which the application will connect. Refer to "Target server" for details.

Argument	Explanation
sslmode	Specify this to encrypt communications. By default, this is disabled. The setting values for sslmode are as follows: disable: Connect without SSL require: Connect always with SSL verify-ca: Connect with SSL, using a certificate issued by a trusted CA (*1) verify-full: Connect with SSL, using a certificate issued by a trusted CA to verify if the server host name matches the certificate (*1)
sslservercertcn	This parameter is enabled only to perform SSL authentication (sslmode=verify-full). Specify the server certificate CN. If this is omitted, the value will be null, and the server certificate CN will be authenticated using the host name specified in host.

*1: If specifying either "verify-ca" or "verify-full", the CA certificate file can be specified using connection string sslrootcert.

When using Driver Manager

Specify the following URL in the API of the DriverManager class:

```
jdbc:postgresql://[host1][:port1],[host2][:port2]/dbName[?targetServerType={primary |
secondary| preferPrimary | preferSecondary | any}][&sslmode=verify-
full&sslrootcert=cACertificateFile&sslservercertcn=targetServerCertificateCN]
```

- If the target server is omitted, the default value "any" is used.
- When using IPV6, specify the host in the "[*host*]" (with square brackets) format.

  [Example]

```
jdbc:postgresql://[2001:Db8::1234]:27500,192.168.1.1:27500/dbName
```

When using the data source

Specify the properties of the data source in the following format:

```
source.setServerName("[host1][:port1],[host2][:port2]");
source.setTargetServerType("primary");
source.setSslmode("verify-full");
source.setSslrootcert("cACertificateFile");
source.setSslservercertcn("targetServerCertificateCN");
```

- If the IP address or host name are omitted, localhost will be used.
- If the port number is omitted, the value specified in the portNumber property will be used. Also, if the portNumber property is omitted, the default is 27500.
- If the target server is omitted, the value will be "any".
- When using IPV6, specify the host in the "[*host*]" (with square brackets) format.

  [Example]

```
source.setServerName("[2001:Db8::1234]:27500,192.168.1.1:27500");
```

Note

If using the connection parameter loginTimeout, the value will be applied for the time taken attempting to connect to all of the specified hosts.

10.2.2 Using the ODBC Driver

Set the following information in the connection string or data source.

Table 10.2 Information to be set

Parameter	Explanation
Servername	Specify IP address 1 and IP address 2, or the host name, using a comma as the delimiter. Based on ODBC rules, it is recommended to enclose the whole string containing comma delimiters with {}. Format: {host1,host2}
Port	Specify the connection destination port numbers, using a comma as the delimiter. Based on ODBC rules, it is recommended to enclose the whole string containing comma delimiters with {}. Format: {port1,port2} Specify the port number corresponding to the IP address or host specified for the nth Servername as the nth Port. The port number can be omitted. If omitted, the default is 27500. If *n* server names are specified, and m ports are specified then there will be error reported. The only exceptions are where m=n or m=1. In case only one port is specified, then the same is applied for all the hosts.
target_session_attrs	Specify the selection sequence of the servers to which the application will connect. Refer to "Target server" for details.
SSLMode	Specify this to encrypt communications. By default, this is disabled. The setting values for SSLMode are as follows: disable: Connect without SSL allow: Connect without SSL, and if it fails, connect with SSL prefer: Connect with SSL, and if it fails, connect without SSL require: Connect always with SSL verify-ca: Connect with SSL, using a certificate issued by a trusted CA (*1) verify-full: Connect with SSL, using a certificate issued by a trusted CA to verify if the server host name matches the certificate (*1)
SSLServerCertCN	This parameter is enabled only to perform SSL authentication (SSLMode=verify-full). Specify the server certificate CN. If this is omitted, the value will be null, and the server certificate CN will be authenticated using the host name specified in Servername.

*1: If specifying either "verify-ca" or "verify-full", use the system environment variable PGSSLROOTCERT of your operating system to specify the CA certificate file as shown below.

Example)
Variable name: PGSSLROOTCERT
Variable value: *cACertificateFile*

When specifying a connection string

Specify the following connection string:

```
...;Servername={host1,host2};Port={port1,port2};[target_session_attrs={any | read-
write | read-only | primary | standby | prefer-standby}];[ SSLMode=verify-
full;SSLServerCertCN=targetServerCertificateCN]...
```

- When using IPV6, specify the host in the "*host*" format.

  [Example]

  ```
  Servername={2001:Db8::1234,192.168.1.1};Port={27500,27500};
  ```

When using the data source

Specify the properties of the data source in the following format:

```
Servername={host1,host2}
Port={port1,port2}
target_session_attrs={any | read-write | read-only | primary | standby | prefer-standby}
SSLMode=verify-full
SSLServerCertCN=targetServerCertificateCN
```

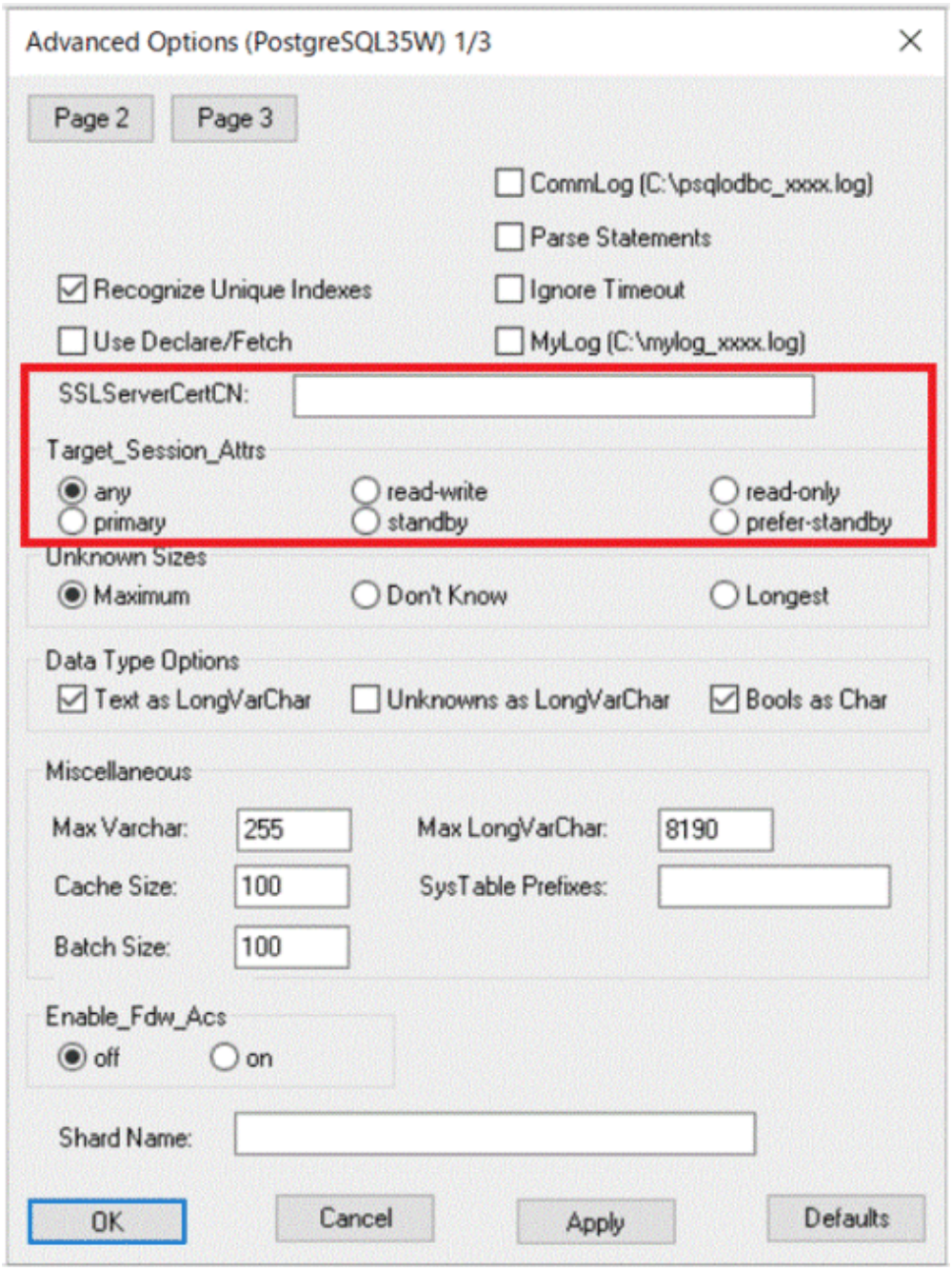
- When using IPV6, specify the host in the "*host*" format.

  [Example]

  ```
  Servername={2001:Db8::1234,192.168.1.1}
  ```

Registering the data source using the ODBC Data Source Administrator

Using the ODBC Data Source Administrator, specify the items within the red border below:



Note

If using the connection parameter login_timeout, this value is applied for connections to each of the specified hosts. If both multiplexed database servers have failed, the connection will time out when a time equal to double the login_timeout value elapses.

10.2.3 Using a .NET Data Provider

Set the following information in the connection string of NpgsqlConnection, or in the data source.

Table 10.3 Information to be set

Argument	Explanation
host1 host2	Specify the IP address or host name.
port1 port2	Specify the port number for the connection.
TargetSessionAttributes	Specify the selection sequence of the servers to which the application will connect. Refer to "Target server" for details.

When specifying a connection string

Specify the following connection string:

```
host1[:port1],host2[:port2];[Target Session Attributes={any | primary | standby |
prefer-primary | prefer-standby | read-write | read-only}];
```

- If the port number is omitted from the host string, the value specified for the Port keyword of the connection string will be used. Refer to "4.3.3 Connection String" for information on the Port keyword.
- When using IPV6, specify the host in the "[*host*]" (with square brackets) format.
- If the target server type is omitted, the value will be any.

  [Example]

```
host=[2001:Db8::1234]:27500,192.168.1.1:27500;
```

When specifying the NpgsqlConnectionStringBuilder property

Specify the Host property of the data source in the following format:

```
host1[:port1],host2[:port2]
```

- If the port number is omitted from the host string, the value specified in the Port property will be used. Also, if the Port property is omitted, the default is 27500.

Specify the TargetSessionAttributes property of the data source in the following format:

```
any | primary | standby | prefer-primary | prefer-standby | read-write | read-only
```

- If the target server type is omitted, the value will be any.

Note

If using the connection parameter Timeout, this value is applied for connections to each of the specified hosts. If both multiplexed database servers have failed, the connection will time out when a time equal to double the Timeout value elapses.

10.2.4 Using a Connection Service File

Set the connection parameters as follows.

Table 10.4 Information to be set

Parameter	Explanation
host	Specify the host names, using a comma as the delimiter.
hostaddr	Specify IP address 1 and IP address 2, using a comma as the delimiter.

Parameter	Explanation
port	Specify the connection destination port numbers, using a comma as the delimiter. Specify the port number for the server specified for the nth host or hostaddr as the nth port. The port number can be omitted. If omitted, the default is 27500. If *n* server names are specified, and m ports are specified then there will be error reported. The only exceptions are where m=n or m=1. In case only one port is specified, then the same is applied for all the hosts.
target_session_attrs	Specify the selection sequence of the servers to which the application will connect. Refer to "Target server" for details.
sslmode	Specify this to encrypt communications. By default, this is disabled. The setting values for sslmode are as follows: disable: Connect without SSL allow: Connect without SSL, and if it fails, connect with SSL prefer: Connect with SSL, and if it fails, connect without SSL require: Connect always with SSL verify-ca: Connect with SSL, using a certificate issued by a trusted CA (*1) verify-full: Connect with SSL, using a certificate issued by a trusted CA to verify if the server host name matches the certificate (*1)
sslservercertcn	This parameter is enabled only to perform SSL authentication (sslmode=verify-full). Specify the server certificate CN. If this is omitted, the value will be null, and the server certificate CN will be authenticated using the host name specified in host.

*1: If specifying either "verify-ca" or "verify-full", use the system environment variable PGSSLROOTCERT (connection parameter sslrootcert) of your operating system to specify the CA certificate file as shown below.

Example)
Variable name: PGSSLROOTCERT
Variable value: *cACertificateFile*

Note

If using the connection parameter connect_timeout, this value is applied for connections to each of the specified hosts. If both multiplexed database servers have failed, the connection will time out when a time equal to double the connect_timeout value elapses.

Point

If using the C Library, embedded SQL or psql commands (including other client commands that specify connection destinations), it is recommended to use a connection service file to specify connection destinations.

In the connection service file, a name (service name) is defined as a set, comprising information such as connection destination information and various types of tuning information set for connections. By using the service name defined in the connection service file when connecting to databases, it is no longer necessary to modify applications when the connection information changes.

10.2.5 Using the C Library (libpq)

It is recommended that you use a connection service file. Refer to "10.2.4 Using a Connection Service File" for details.

If a connection service file will not be used, set the following information for the database connection control functions (PQconnectdbParams, PQconnectdb, and so on) or environment variables.

Table 10.5 Information to be set

Parameter (environment variable name)	Explanation
host(PGHOST)	Specify the host names, using a comma as the delimiter.
hostaddr(PGHOSTADDR)	Specify IP address 1 and IP address 2, using a comma as the delimiter.
port(PGPORT)	Specify the connection destination port numbers, using a comma as the delimiter. Specify the port number for the server specified for the nth host or hostaddr as the nth port. The port number can be omitted. If omitted, the default is 27500. If *n* server names are specified, and m ports are specified then there will be error reported. The only exceptions are where m=n or m=1. In case only one port is specified, then the same is applied for all the hosts.
target_session_attrs(PGTARGETSESSIONATTRS)	Specify the selection sequence of the servers to which the application will connect. Refer to "Target server" for details.
sslmode(PGSSLMODE)	Specify this to encrypt communications. By default, this is disabled. The setting values for sslmode are as follows: disable: Connect without SSL allow: Connect without SSL, and if it fails, connect with SSL prefer: Connect with SSL, and if it fails, connect without SSL require: Connect always with SSL verify-ca: Connect with SSL, using a certificate issued by a trusted CA (*1) verify-full: Connect with SSL, using a certificate issued by a trusted CA to verify if the server host name matches the certificate (*1)
sslservercertcn(PGXSSLSERVERCERTCN)	This parameter is enabled only to perform SSL authentication (sslmode=verify-full). Specify the server certificate CN. If this is omitted, the value will be null, and the server certificate CN will be authenticated using the host name specified in host.

*1: If specifying either "verify-ca" or "verify-full", use the system environment variable PGSSLROOTCERT (connection parameter sslrootcert) of your operating system to specify the CA certificate file as shown below.

Example)
Variable name: PGSSLROOTCERT
Variable value: *cACertificateFile*

When using URI

```
postgresql://host1[:port1],host2[:port2][,...]/database_name
[?target_session_attrs={read-write | read-only | primary | standby | prefer-standby |
any }]
```

- When using IPV6, specify the host in the "[*host*]" (with square brackets) format.

  [Example]

```
postgresql://postgres@[2001:Db8::1234]:27500,192.168.1.1:27500/database_name
```

When using key-value

```
host=host1[,host2] port=port1[,port2] user=user1 password=pwd1 dbname=mydb
[target_session_attrs={read-write | read-only | primary | standby | prefer-standby |
any }]
```

- When using IPV6, specify the host in the "*host*" format.

  [Example]

```
host=2001:Db8::1234,192.168.1.1 port=27500,27500
```

Note

If using the connection parameter connect_timeout, this value is applied for connections to each of the specified hosts. If both multiplexed database servers have failed, the connection will time out when a time equal to double the connect_timeout value elapses.

Information

If using a password file (.pgpass), describe the entries matching each server.

- Example 1:

```
host1:port1:dbname:user:password
host2:port2:dbname:user:password
```

- Example 2:

```
*:port:dbname:user:password
```

10.2.6 Using Embedded SQL

It is recommended that you use a connection service file. Refer to "10.2.4 Using a Connection Service File" for details.

Point

If using a connection service file, either of the following methods is available:

- Set the service name as a string literal or host variable, as follows:

  tcp:postgresql://?service=my_service

- Set the service name in the environment variable PGSERVICE, and use CONNECT TO DEFAULT

If a connection service file will not be used, use a literal or variable to specify the connection destination server information for target in the SQL statement below:

```
EXEC SQL CONNECT TO target [AS connection-name] [USER user-name];
```

Method used

```
dbname@host1,host2[:[port1][,port2]]
tcp:postgresql://host1,host2[:[port1][,port2]] [/dbname] [?target_session_attrs={read-
write | read-only | primary | standby | prefer-standby | any}][&sslmode=verify-
full&sslservercertcn=targetServerCertificateCN]
```

- The above format cannot be specified directly without using a literal or variable.

Table 10.6 Information to be set

Argument	Explanation
host1 host2	Specify the IP address or host name. IPv6 format addresses cannot be specified.
port1 port2	Specify the connection destination port numbers, using a comma as the delimiter. The port number can be omitted. If omitted, the default is 27500.
dbname	Specify the database name.
target_session_attrs	Specify the selection sequence of the servers to which the application will connect. Refer to "Target server" for details.
sslmode	Specify this to encrypt communications. By default, this is disabled. The setting values for sslmode are as follows: disable: Connect without SSL allow: Connect without SSL, and if it fails, connect with SSL prefer: Connect with SSL, and if it fails, connect without SSL require: Connect always with SSL verify-ca: Connect with SSL, using a certificate issued by a trusted CA (*1) verify-full: Connect with SSL, using a certificate issued by a trusted CA to verify if the server host name matches the certificate (*1)
sslservercertcn	This parameter is enabled only to perform SSL authentication (sslmode=verify-full). Specify the server certificate CN. If this is omitted, the value will be null, and the server certificate CN will be authenticated using the host name specified in host.

*1: If specifying either "verify-ca" or "verify-full", use the system environment variable PGSSLROOTCERT (connection parameter sslrootcert) of your operating system to specify the CA certificate file as shown below.

Example)
Variable name: PGSSLROOTCERT
Variable value: *cACertificateFile*

Point

Environment variables can also be used. Refer to "10.2.5 Using the C Library (libpq)" for information on environment variables.

Note

If using the connection parameter connect_timeout, this value is applied for connections to each of the specified hosts. If both multiplexed database servers have failed, the connection will time out when a time equal to double the connect_timeout value elapses.

10.2.7 Using the psql Command

It is recommended that you use a connection service file. Refer to "10.2.4 Using a Connection Service File" for details.

If a connection service file will not be used, specify the following information in the psql command option/environment variable.

Table 10.7 Information to be set

Option (environment variable)	Explanation
-h/--host(PGHOST/ PGHOSTADDR)	Specify IP address 1 and IP address 2, or the host name, using a comma as the delimiter. This can also be specified for the environment variable PGHOST or PGHOSTADDR.
-p/--port(PGPORT)	Specify the connection destination port numbers, using a comma as the delimiter. This can also be specified for the environment variable PGPORT. Specify the port number corresponding to the IP address specified for the nth -h option as the nth -p option. The port number can be omitted. If omitted, the default is 27500. If *n* -h options are specified, and m -p options are specified then there will be error reported. The only exception is where m=n or m=1. In case only one port is specified, then the same is applied for all the hosts.
(PGTARGETSESSIONATTRS)	Specify the selection sequence of the servers to which the application will connect. Refer to "Target server" for details.
(PGSSLMODE)	Specify this to encrypt communications. By default, this is disabled. The setting values for PGSSLMODE are as follows: disable: Connect without SSL allow: Connect without SSL, and if it fails, connect with SSL prefer: Connect with SSL, and if it fails, connect without SSL require: Connect always with SSL verify-ca: Connect with SSL, using a certificate issued by a trusted CA (*1) verify-full: Connect with SSL, using a certificate issued by a trusted CA to verify if the server host name matches the certificate (*1)
(PGXSSLSERVERCERTCN)	This environment variable is enabled only to perform SSL authentication (PGSSLMODE=verify-full). Specify the server certificate CN. If this is omitted, the value will be null, and the server certificate CN will be authenticated using the host name specified in host.

*1: If specifying either "verify-ca" or "verify-full", use the system environment variable PGSSLROOTCERT (connection parameter sslrootcert) of your operating system to specify the CA certificate file as shown below.

Example)
Variable name: PGSSLROOTCERT
Variable value: *cACertificateFile*

Note

If using the connection parameter connect_timeout, this value is applied for connections to each of the specified hosts. If both multiplexed database servers have failed, the connection will time out when a time equal to double the connect_timeout value elapses.

Information

Use the same method as for psql commands to specify connection destination server information for other client commands used to specify connection destinations.

Chapter 11 Performance Tuning

This chapter explains how to tune application performance.

11.1 Enhanced Query Plan Stability

Fujitsu Enterprise Postgres estimates the cost of query plans based on SQL statements and database statistical information, and selects the least expensive query plan. However, like other databases, Fujitsu Enterprise Postgres does not necessarily select the most suitable query plan. For example, it may suddenly select unsuitable query plan due to changes in the data conditions.

In mission-critical systems, stable performance is more important than improved performance, and changes in query plans case to be avoided. In this situation, by stabilizing the SQL statement query plan so that it does not change, deterioration of the application performance is suppressed.

11.1.1 Optimizer Hints

This section explains the basic feature content of the optimizer hint (pg_hint_plan).

Refer to the open-source software webpage for information on pg_hint_plan.

In Fujitsu Enterprise Postgres, the optimizer hints can be specified in all application interfaces.

Description

You can specify a query plan in each SQL statement.

List of Features

The main query plans that can be specified using this feature are as follows:

- Query methods
- Join methods
- Join sequences

Query methods

Specify which method to use to query the specified table.

The main features are as follows:

- SeqScan (*tableName*)
- BitMapScan (*tableName* [*indexName* ...])
- IndexScan (*tableName* [*indexName* ...])
- IndexOnlyScan (*tableName* [*indexName* ...])

Note

- If the specified index does not exist, or is not related to the search condition column specified in the WHERE clause, for example, SeqScan will be used.
- Even if IndexOnlyScan is specified, IndexScan may be used if it is necessary to access the table because a row was updated, for example.
- If multiple query methods were specified for the same table, the method specified last will be used.

Join methods

Specify the join method.

The main features are as follows:

- NestLoop (*tableName tableName* [*tableName* ...])
- MergeJoin (*tableName tableName* [*tableName* ...])
- HashJoin (*tableName tableName* [*tableName* ...])

- These cannot be specified for view tables and subqueries.
- If multiple methods were specified for the same table combination, the method specified last will be used.

Join sequences

The tables will be joined in the specified table sequence.

Specify the information using the following method:

- Leading ((*table table*))

  The method used to specify [*table*] is as follows:

  table = *tableName* or (*table table*)

If multiple sequences were specified for the same table combination, the sequence specified last will be used.

Usage method

The use of this feature is explained below.

Method used to define this feature

Define this feature by specifying the format (block comment) " /*+ ... */".

- To specify hint clauses in each SELECT statement, for example when there are multiple SELECT statements in the SQL statement, define all hint clauses in the first block comment.

Specifying hint clauses for the emp table and the dept table

```
WITH /*+ IndexScan(emp emp_age_index) IndexScan(dept dept_deptno_index) */ age30
AS (SELECT * FROM emp WHERE age BETWEEN 30 AND 39)
SELECT * FROM age30, dept WHERE age30.deptno = dept.deptno;
```

- To specify separate hint clauses for the same object in the SQL statement, define aliases in each object, and then specify hint clauses for those aliases.

Specifying separate hint clauses for the emp table

```
WITH /*+ SeqScan(ta) IndexScan(tb) */ over100
AS (SELECT empno FROM emp ta WHERE salary > 1000000)
SELECT * FROM emp tb, over100 WHERE tb.empno = over100.empno AND tb.age < 30
```

- When using embedded SQL in C, the locations in which the hint clause block comment is specified are restricted. Refer to "6.4.2 Compiling Applications" for details.

Usage notes

- If a hint clause was specified in multiple block comments in the SQL statement, the hint clause specified in the second block comment and thereafter will be ignored.
- If characters other than those listed below appear before the hint clause in the SQL statement, they will be invalid even for hint clause block comments.
  - Space, tab, line feed
  - Letter (uppercase and lowercase), number
  - Underscore, comma
  - Brackets ()

11.1.2 Locked Statistics

This section explains the basic feature content for locked statistics (pg_dbms_stats).

Refer to the open-source software webpage for information on pg_dbms_stats.

Description

Locks the statistics.

By using this feature to lock the statistics for performance obtained in job load testing in an environment that simulates a production environment, performance degradation caused by changes to the query plan after go-live can be suppressed.

Additionally, by using the export and import features, statistics that were checked in the test environment can also be reproduced in the production environment.

List of Features

The main features that can be specified using this feature are as follows.

[Features]

Feature	Details	Description
Lock/unlock of the statistics	Lock	Locks the statistics so that the currently selected query plan remains selected.
	Unlock	Unlocks the statistics.
Backup/restore of the statistics	Backup	Backs up the current statistics.
	Restore	Restores the statistics to the point when they were backed up, and then locks them.
	Purge	Deletes backups that are no longer necessary.
Backup/restore using external files	Export	Outputs the current statistics to an external file (binary format).
	Import	Reads the statistics from an external file created by the export feature, and then locks them.

[Target object]

Target resource	Range of feature
Database	In the database
Schema	In the schema
Table	In the table
Column	ID column

Usage method

The use of this feature is explained below.

Method used to specify this feature

Specify this feature as an SQL function.

The methods used to specify the main features are shown in the table below.

Feature	Object	Function specified
Lock	Database	dbms_stats.lock_database_stats()
	Schema	dbms_stats.lock_schema_stats('*schemaName*')
	Table	dbms_stats.lock_table_stats('*schemaName.tableName*')
Unlock	Database	dbms_stats.unlock_database_stats()
	Schema	dbms_stats.unlock_schema_stats('*schemaName*')
	Table	dbms_stats.unlock_table_stats('*schemaName.tableName*')
Import	Database	dbms_stats.import_database_stats('*fullPathOfExportedFile*')
Backup	Database	dbms_stats.backup_database_stats('*commentUsedForIdentification*')
Restore	Database	[Format 1] dbms_stats.restore_database_stats('*timestamp*') [Timestamp] Specify in the same format as the time column of the backup_history table. Backups earlier than the specified time will be restored. [Format 2] dbms_stats.restore_stats(*backupId*) [Backup ID] Specify a value in the id column of the backup_history table. The specified backup will be restored.
Purge	Backup	dbms_stats.purge_stats(*backupId,flagUsedForDeletion*) [Backup ID] Specify a value in the id column of the backup_history table. [Flag used for deletion] true: The target backup is forcibly deleted. false: The target backup is deleted only when there are also backups for the entire database.

Remark 1: The export feature is executed using the COPY statement, not the SQL function.

Example

Example 1: Locking the statistics of the entire database

```
userdb=# SELECT dbms_stats.lock_database_stats();
  lock_database_stats
----------------------
  tbl1
  tbl1_pkey
```

Note that the locked information can be referenced as follows:

```
userdb=# select relname  from dbms_stats.relation_stats_locked;
  relname
----------------------
  tbl1
  tbl1_pkey
```

Example 2: Unlocking the statistics of the entire database

```
userdb=# SELECT dbms_stats.unlock_database_stats();
  unlock_database_stats
----------------------
  tbl1
  tbl1_pkey
```

Example 3: Backing up the statistics of the entire database

```
userdb=# SELECT dbms_stats.backup_database_stats('backup1');
  backup_database_stats
----------------------
                     1
```

Note that the backed up statistics can be referenced as follows:

```
userdb=# select id,comment,time,unit from dbms_stats.backup_history;
  id | comment  |             time              | unit
 ----+----------+-------------------------------+------
   1 | backup1  | 2014-03-04 11:08:40.315948+09 | d
```

The ID:1 backup "backup1" is obtained for each database at "2014-03-04 11:08:40.315948+09".
[Meaning of unit] d: database s: schema t: table c: column

Example 4: Exporting the statistics of the entire database

```
$ psql -d userdb -f export.sql
BEGIN
COMMIT
```

export.sql is the file in which the COPY statement is defined.
Refer to "export_effective_stats-<*x*>.sql_sample" for information on the content of the COPY statement. "<*x*>" indicates the product version.

"export_effective_stats-<*x*>.sql_sample" is stored as follows:
fujitsuEnterprisePostgresInstallDir/share/doc/extension

Example 5: Importing the statistics of the entire database

```
$ psql -d userdb -c  "SELECT dbms_stats.import_database_stats ('$PWD/
export_stats.dmp')"
 import_database_stats
-----------------------

(1 row)
```

Usage notes

- You must run the ANALYZE command once for the target tables of this feature. If the ANALYZE command is not run, the statistics cannot be locked.
  Refer to "SQL Commands" in "Reference" in the PostgreSQL Documentation for information on the ANALYZE command.
- To use this feature to delete an object that has locked the statistics, use the unlock feature to delete the object lock information first.
- This feature does not specify the statistics value directly. It reproduces the status that has actually occurred. For this reason, if the text format is specified in the COPY statement when the export occurs, restore will not be possible. Always use the binary format when performing the export.

Chapter 12 Scan Using a Vertical Clustered Index (VCI)

This chapter describes scanning using a VCI.

Note

This feature can only be used in Advanced Edition.

12.1 Operating Conditions

Faster aggregation can be achieved by using a VCI defined for all columns to be referenced.

This section describes the conditions under which a scan can use a VCI.

Whether to use VCI is determined based on cost estimation in the same way as normal indexes. Therefore, another execution plan will be selected if it is cheaper than a VCI even if a VCI is available.

SQL statements that can use VCIs

In addition to general SELECT statements, VCIs can be used for the SQL statements below (as long as they do not specify any of the elements listed in "SQL statements that cannot use VCIs" below):

- SELECT INTO
- CREATE TABLE AS SELECT
- CREATE MATERIALIZED VIEW ... AS SELECT
- CREATE VIEW ... AS SELECT
- COPY (SELECT ...) TO

SQL statements that cannot use VCIs

VCIs cannot be used for SQL statements that specify any of the following:

- Subquery to reference the column in which the parent query is referencing is specified
- Lock clause (such as FOR UPDATE)
- Cursor declared with WITH HOLD or scrollable
- SERIALIZABLE transaction isolation level
- Function or operator listed in "Functions and operators that do not use a VCI"
- User-defined function

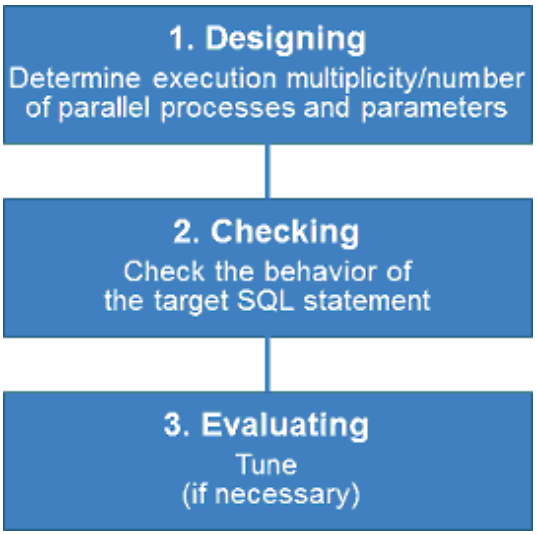
Table 12.1 Functions and operators that cannot use VCIs

Classification		Function/operator
Mathematical functions and operators	Random functions	random and setseed
String functions and operators	String functions	format (if the *format* argument is specified), regexp_matches, regexp_split_to_array and regexp_split_to_table
Date/time functions and operators	Date/time functions	age(timestamp), current_date, current_time, current_timestamp, localtime, localtimestamp, statement_timestamp and transaction_timestamp
	Delaying execution functions	pg_sleep, pg_sleep_for, and pg_sleep_until

Classification		Function/operator
Enum support functions		All functions and operators
Geometric functions and operators		All functions and operators
Network address functions and operators		All functions and operators
Text search functions and operators		All functions and operators
XML functions		All functions
JSON functions and operators		All functions and operators
Sequence manipulation functions		All functions
Array functions and operators		All functions and operators
Range functions and operators		All functions and operators
Aggregate functions	General-purpose aggregate functions	array_agg, json_agg, json_object_agg, string_agg and xmlagg
	Aggregate functions for statistics	corr, covar_pop, covar_samp, regr_avgx, regr_avgy, regr_count, regr_intercept, regr_r2, regr_slope, regr_sxx, regr_sxy and regr_syy
	Ordered-set aggregate functions	All functions
	Hypothetical-set aggregate functions	All functions
Window functions		All functions
Subquery expressions		Subquery expressions with its row constructor specified on the left side
Row and array comparisons		Row constructor and composite type comparisons
Set returning functions		All functions
System information functions		All functions
System administration functions		All functions
Trigger functions		All functions
Session information functions		current_role and current_user

12.2 Usage

This section describes how to use a VCI in line with the following steps:



12.2.1 Designing

Design as follows before using a VCI.

- Execution multiplicity and number of parallel processes
- Parameters

Execution multiplicity and number of parallel processes

Determine the maximum number of SQL statements that can be executed simultaneously and the number of parallel processes based on the number of CPU cores that can be allocated for scans that use VCI to perform aggregate processing. Design in advance the multiplicity of SQL statements for executing scans that use VCI and the number of parallel processes for scans that use VCI.

For example, if the number of CPUs that can be allocated is 32 cores, then the maximum number of SQL statements that can be executed simultaneously is 8 and the number of parallel processes is 4.

Note

L

A temporary file is created in /dev/shm or in a directory specified for the vci.smc_directory parameter as the dynamic shared memory for each SQL statement during a scan using a VCI.

W

A temporary file is created in a directory under the data storage directory or in a directory specified for the vci.smc_directory parameter as the dynamic shared memory for each SQL statement during a scan using a VCI.

Ensure that this directory has sufficient space to meet the memory requirements estimated for the execution multiplicity and number of parallel processes of SQL statements (refer to "Memory used per scanning" in "VCI Memory Requirements" in the Installation and Setup Guide for Server for details). If it does not have sufficient space when a scan is performed, SQL statements will return errors due to the insufficient memory.

Parameters

The VCI parallel scan feature cannot be used for setting parameters immediately after creating an instance.

Therefore, set the parameters below based on the values determined in "Execution multiplicity and number of parallel processes of SQL statements" above.

Parameter name	Description	Default	Value index
vci.max_parallel_degree	Maximum number of VCI parallel processes	0	Specify the number of parallel processes.

Parameter name	Description	Default	Value index
	(background processes) to be used per SQL statement.		
vci.smc_directory	Directory name in which a temporary file is created as the dynamic shared memory during a scan using a VCI.	L /dev/shm W A directory (*dataStorageDir*\\base\\pgsql_tmp) under the data storage directory	Specify a directory that has enough free space for the memory used for each query during the scan.
max_worker_processes	Maximum number of background processes that the system supports.	8	Add the value of the maximum number of SQL statements that can be executed simultaneously for scans that use VCI multiplied by vci.max_parallel_degree.

See

Refer to "Parameters" in the Operation Guide for information on the details of and how to set the parameters.

12.2.2 Checking

Execute the SQL statement with "EXPLAIN ANALYZE" to check the following:

- If a VCI was used
  "Custom Scan (VCI...)" is displayed in the plan if a VCI was used.
- Number of parallel processes
  The number of parallel processes when the SQL statement is executed is displayed in "Allocated Workers". Check that it is running the designed number of parallel processes.
- Response
  Check if the execution time displayed in "Execution time" is as estimated.

The following shows an example of the output result of EXPLAIN ANALYZE:

```
EXPLAIN ANALYZE SELECT COUNT(*) FROM test WHERE x > 10000;
                                                    QUERY
PLAN
-----------------------------------------------------------------------------------------
-------------------------
Custom Scan (VCI Aggregate)  (cost=19403.15..19403.16 rows=1 width=0) (actual
time=58.505..58.506 rows=1 loops=1)
       Allocated Workers: 4
   ->  Custom Scan (VCI Scan) using test_x_idx on test (cost=0.00..16925.00 rows=991261
width=0) (never executed)
         Filter: (x > 10000)
Planning time: 0.151 ms
Execution time: 86.910 ms
(6 rows)
```

Note

A cost output by the execution plan that uses a VCI may be inaccurate. A VCI works if all or part of the best execution plan when the SQL statement was executed is replaced with an execution plan that uses a VCI. If the cost of the execution plan to be replaced is lower than a certain value (vci.cost_threshold parameter), it will not be replaced or recalculated. Therefore, the cost of the original execution plan is output as is.

12.2.3 Evaluating

If the results in "12.2.2 Checking" is any of the following, tune accordingly:

If a VCI is not used

- Check if the "12.1 Operating Conditions" are met.
- Check if vci.enable is set to "on".
- A VCI may not be appropriately used when statistics are outdated, such as immediately after inserting a large amount of data. In such cases, execute the VACUUM ANALYZE statement or the ANALYZE statement.
- A VCI is not used if there is insufficient memory for VCI scan. This may occur during time-consuming transactions involving tables for which VCIs were defined. Set vci.log_query to "on", and check if either "could not use VCI: local ROS size (%zu) exceeds limit (%zu)" or "out of memory during local ROS generation" is output. If it is, then increase the value of the vci.max_local_ros.

Response is not as expected

Tuning may improve response. Check the following:

- If vci.max_parallel_degree is not set or is set to 0, set an appropriate value according to "12.2.1 Designing".
- If there is a margin in the CPU usage, increase the value of vci.max_parallel_degree and check again. In addition, if the value that of max_worker_processes is lower than the maximum number of SQL statements that can be executed simultaneously for parallel scan multiplied by vci.max_parallel_degree, increase it and check again.

12.3 Usage Notes

This section provides notes on using VCI.

- Regardless of whether VCI is used, the content of the result does not change. However, records may be returned in a different order if the ORDER BY clause is not specified.
- To reduce resource consumption, edit postgresql.conf or use the SET statement to enable/disable vci.enable when you use this feature only for specific times or jobs (SQL applications).
- The optimizer hint (pg_hint_plan) cannot be specified for a VCI. The hint clause is ignored if it is specified.
- If a plan other than VCI is specified for the optimizer hint (pg_hint_plan), a VCI may be used. Therefore, if you specify a query plan with the hint clause, use the SET statement to set vci.enable to "off".
- The message below may be output when a scan that uses VCI is performed on the streaming replication standby server:

```
"LOG: recovery has paused"
"HINT: Execute pg_wal_replay_resume() to continue."
```

  This message is output because application of the WAL to the VCI temporarily pauses due to the scan being performed.

- Even if a scan is performed using a VCI, information in the idx_scan, idx_tup_read, and idx_tup_fetch columns of the collected statistics views, pg_stat_all_indexes and pg_stat_user_indexes, will not be updated.

- Currently, it is not possible to replace the query plan for parallel aggregation with the query plan using VCI. Therefore, if you create a VCI on a column of a partition table and aggregate (sum () etc.) on that column, one of the following plans will be selected. Use different setting parameters according to the situation of the target table.

  - Plan of the parallel aggregations using scan methods other than VCI scan

    It is selected when max_parallel_workers_per_gather is 1 or more.

```
explain select sum(value) from test;
                                          QUERY PLAN
-----------------------------------------------------------------------------------------
 Finalize Aggregate  (cost=99906.30..99906.31 rows=1 width=8)
   ->  Gather  (cost=99906.08..99906.29 rows=2 width=8)
         Workers Planned: 2
         ->  Partial Aggregate  (cost=98906.08..98906.09 rows=1 width=8)
               ->  Parallel Append  (cost=0.00..94739.83 rows=1666500 width=4)
                   ->  Parallel Seq Scan on test_1  (cost=0.00..43203.67 rows=833250
width=4)
                   ->  Parallel Seq Scan on test_2  (cost=0.00..43203.67 rows=833250
width=4)
```

    This plan is fast when the number of records to be aggregated (number of records that hit the search conditions) is very large. This is because the benefit of parallelizing aggregation is important, not the performance of scanning. For example, each parallel worker will perform a sequential scan and aggregate most of the scanned records.

  - Plan that aggregates VCI scan results by a single aggregator node

    It is selected by setting max_parallel_workers_per_gather to 0 and not creating a query plan of parallel aggregate.

```
explain select sum(value) from test;
                                       QUERY PLAN
-----------------------------------------------------------------------------------------
 Aggregate  (cost=145571.00..145571.01 rows=1 width=8)
   ->  Append  (cost=0.00..135572.00 rows=3999600 width=4)
         ->  Custom Scan (VCI Scan) using test_1_id_value_idx on test_1
(cost=0.00..57787.00 rows=1999800 width=4)
               Allocated Workers: 2
         ->  Custom Scan (VCI Scan) using test_2_id_value_idx on test_2
(cost=0.00..57787.00 rows=1999800 width=4)
               Allocated Workers: 2
```

    This plan is fast when the number of aggregated items is not large or when the size of the aggregated column is smaller than the record size. This is because the scan performance is more important, so it is faster to aggregate the results of VCI scans of each partition.

- Originally, if there is only one partition to be accessed, the following VCI aggregation plan can be used. Below is an example of scanning only one partition with partition pruning.

```
explain select sum(value) from test where id < 1000001;
                                       QUERY PLAN
-----------------------------------------------------------------------------------------
 Custom Scan (VCI Aggregate)  (cost=62786.50..62786.51 rows=1 width=8)
   Allocated Workers: 2
   ->  Custom Scan (VCI Scan) using test_1_id_value_idx on test_1  (cost=0.00..57787.00
rows=1999800 width=4)
         Filter: (id < 1000001)
```

  However, the current planner does not try to choose VCI aggregation because it creates a plan for parallel aggregation if the table is partitioned. So in this case, set max_parallel_workers_per_gather to 0 to force the planner to choose VCI aggregation.

Appendix A Precautions when Developing Applications

This appendix describes precautions when developing applications with Fujitsu Enterprise Postgres.

A.1 Precautions when Using Functions and Operators

This section describes notes for using functions and operators.

A.1.1 General rules of Functions and Operators

This section describes general rules for using functions and operators. Ensure the general rules are followed when using functions and operators to develop applications.

General rules

- Specify the stated numbers for arguments when specifying numbers for arguments in functions.
- Specify the stated data types when specifying data types for functions. If you use a data type other than the stated data types, use CAST to explicitly convert the data type.
- Specify data types that can be compared when specifying data types for operators. If you use a data type that cannot be compared, use CAST to explicitly convert the data type.

See

Refer to "Functions and Operators" under "The SQL Language" in the PostgreSQL Documentation for information on the functions and operators available with Fujitsu Enterprise Postgres.

A.1.2 Errors when Developing Applications that Use Functions and/or Operators

This section provides examples of problems that may occur when developing applications that use functions and/or operators, and describes how to deal with them.

The error "Function ***** does not exist" occurs when executing SQL

The following error will occur when executing an SQL statement that does not abide by the general rules for functions:

```
ERROR:  Function ****** does not exist
```

Note: "*****" denotes the function for which the error occurred, and the data type of its arguments.

The cause of the error will be one of the following:

- The specified function does not exist.
- The wrong number of arguments or wrong argument data type was specified

Corrective action

Check the following points and correct any errors:

- Check if there are any errors in the specified function name, number of arguments, or argument data type, and revise accordingly.
- Check the argument data type of the function displayed in the message. If an unintended data type is displayed, use a function such as CAST to convert it.

The error "Operator does not exist" occurs when executing SQL

The following error will occur when executing an SQL statement that specifies a data type in the operator that cannot be compared:

```
ERROR:  Operator does not exist: *****
```

Note: "*****" denotes the operator for which the error occurred, and the data type of the specified value.

Corrective action

Ensure the data type of the expressions specified on the left and right sides of the operator can be compared. If required, revise to ensure these data types can be compared by using a function such as CAST to explicitly convert them.

A.2 Notes when Using Temporary Tables

In standard SQL, a temporary table can be defined in advance to enable an empty temporary table to be created automatically when the application connects to the database. However, in Fujitsu Enterprise Postgres, a temporary table must be created when the application connects to the database by explicitly using the CREATE TABLE statement.

If the same temporary table is repeatedly created and deleted during the same session, the system table might expand, and memory usage might increase. To prevent this, specify the CREATE TABLE statement to ensure the temporary table is reused.

For example, in cases where a temporary table would be created and deleted for repeatedly executed transactions, specify the CREATE TABLE statement as shown below:

- Specify "IF NOT EXISTS" to create a temporary table only if none exists when the transaction starts.
- Specify "ON COMMIT DELETE ROWS" to ensure all rows are deleted when the transaction ends.

See

Refer to "SQL Commands" under "Reference" in the PostgreSQL Documentation for information on the CREATE TABLE statement.

Examples of SQL using a temporary table are shown below:

Example of bad use (creating and deleting a temporary table)

```
BEGIN;
CREATE TEMPORARY TABLE mytable(col1 CHAR(4), col2 INTEGER) ON COMMIT DROP;
     (mytable processes)

COMMIT;
```

Example of good use (reusing a temporary table)

```
BEGIN;
CREATE TEMPORARY TABLE IF NOT EXISTS mytable(col1 CHAR(4), col2 INTEGER) ON COMMIT
DELETE ROWS;
     (mytable processes)

COMMIT;
```

A.3 Implicit Data Type Conversions

An implicit data type conversion refers to a data type conversion performed automatically by Fujitsu Enterprise Postgres, without the need to explicitly specify the data type to convert to.

The combination of possible data type conversions differs, depending on whether the expression in the conversion source is a literal.

For non-literals, data types can only be converted to other types within the same range.

For literals, character string literal types can be converted to the target data type. Numeric literals are implicitly converted to specific numeric types. These implicitly converted numeric literals can then have their types converted to match the conversion target data type within the numeric type range. For bit character string literals, only the bit column data type can be specified. The following shows the range of type conversions for literals.

Table A.1 Data type combinations that contain literals and can be converted implicitly

Conversion target		Conversion source		
		Character literal (*1)	Numeric literal(*2)	Bit character string literal
Numeric type	SMALLINT	Y	N	N
	INTEGER	Y	Y (*3)	N
	BIGINT	Y	Y (*4)	N
	DECIMAL	Y	Y (*5)	N
	NUMERIC	Y	Y (*5)	N
	REAL	Y	N	N
	DOUBLE PRECISION	Y	N	N
	SMALLSERIAL	Y	N	N
	SERIAL	Y	Y (*3)	N
	BIGSERIAL	Y	Y (*4)	N
Currency type	MONEY	Y	N	N
Character type	CHAR	Y	N	N
	VARCHAR	Y	N	N
	NCHAR	Y	N	N
	NCHAR VARYING	Y	N	N
	TEXT	Y	N	N
Binary data type	BYTEA	Y	N	N
Date/time type	TIMESTAMP WITHOUT TIME ZONE	Y	N	N
	TIMESTAMP WITH TIME ZONE	Y	N	N
	DATE	Y	N	N
	TIME WITHOUT TIME ZONE	Y	N	N
	TIME WITH TIME ZONE	Y	N	N
	INTERVAL	Y	N	N
Boolean type	BOOLEAN	Y	N	N
Geometric type	POINT	Y	N	N
	LSEG	Y	N	N
	BOX	Y	N	N

Conversion target		Conversion source		
		Character literal (*1)	Numeric literal(*2)	Bit character string literal
	PATH	Y	N	N
	POLYGON	Y	N	N
	CIRCLE	Y	N	N
Network address type	CIDR	Y	N	N
	INET	Y	N	N
	MACADDR	Y	N	N
	MACADDR8	Y	N	N
Bit string type	BIT	Y	N	Y
	BIT VARYING	Y	N	Y
Text search type	TSVECTOR	Y	N	N
	TSQUERY	Y	N	N
UUID type	UUID	Y	N	N
XML type	XML	Y	N	N
JSON type	JSON	Y	N	N

Y: Can be converted
N: Cannot be converted

*1: Only strings that can be converted to the data type of the conversion target can be specified (such as "1" if the conversion target is a numeric type)

*2: "Y" indicates specific numeric types that are converted first.

*3: Integers that can be expressed as INTEGER types can be specified

*4: Integers that cannot be expressed as INTEGER types, but can be expressed as BIGINT types, can be specified

*5: Integers that cannot be expressed as INTEGER or BIGINT types, but that can be expressed as NUMERIC types, or numeric literals that contain a decimal point or the exponent symbol (e), can be specified

Implicit data type conversions can be used when comparing or storing data.

The conversion rules differ, depending on the reason for converting. Purpose-specific explanations are provided below.

A.3.1 Function Argument

Value expressions specified in a function argument will be converted to the data type of that function argument.

See

Refer to "Functions and Operators" under "The SQL Language" in the PostgreSQL Documentation for information on data types that can be specified in function arguments.

A.3.2 Operators

Comparison operators, BETWEEN, IN

Combinations of data types that can be compared using comparison operators, BETWEEN, or IN are shown below.

Table A.2 Combinations of comparable data type

Left side	Right side		
	Numeric type	Character string type	Date/time type
Numeric type	Y	N	N
Character type	N	Y	N
Date/time type	N	N	Y

Y: Can be compared
N: Cannot be compared

When strings with different lengths are compared, the shorter one is padded with spaces to make the lengths match.

When numeric values with different precisions are compared, data will be converted to the type with the higher precision.

Set operation and CASE also follow the same rules.

Other operators

Value expressions specified in operators will be converted to data types that are valid for that operator.

See

Refer to "Functions and Operators" under "The SQL Language" in the PostgreSQL Documentation for information on data types that can be specified in operators.

A.3.3 Storing Values

Value expressions specified in the VALUES clause of the INSERT statement or the SET clause of the UPDATE statement will be converted to the data type of the column in which they will be stored.

A.4 Notes on Using Index

This section explains the notes on using the following indexes:

- SP-GiST index

A.4.1 SP-GiST Index

If more than 2 concurrent updates are performed on a table in which the SP-GiST index is defined, applications may stop responding. When this occur, all system processes including the Check Pointer process will also be in the state of no response. For these reasons, use of the SP-GiST index is not recommended.

A.5 Notes on Using Multibyte Characters in Definition Names

Do not use multibyte characters in database names or user names if using a Windows database server.

Multibyte characters must not be used in database names or user names on non-Windows database servers, because certain conditions may apply or it may not be possible to connect to some clients.

Related notes and constraints are described below.

1) Configuring the client encoding system

The client encoding system must be configured when the names are created.

See

Refer to "Character Set Support" in "Server Administration" in the PostgreSQL Documentation for information on how to configure the client encoding system.

2) Encoding system of names used for connection

Ensure that the encoding system of names used for connection is the same as that of the database that was connected when these names were created.

The reasons for this are as follows:

- Storage system for names in Fujitsu Enterprise Postgres

  The system catalog saves encoded names by using the encoding system of the database at the time the names were created.

- Encoding conversion policy when connected

  When connected, names sent from the client are matched with names in the system catalog without performing encoding conversion.

Accordingly, if the database that was connected when the names were defined uses the EUC_JP encoding system, but the database name is specified using UTF-8 encoding, then the database will be considered to be non-existent.

3) Connection constraints

The table below shows the connection constraints for each client type, based on the following assumptions:

- The conditions described in 1) and 2) above are satisfied.
- The database name and user names use the same encoding system.

Client type	Client operating system	
	Windows(R)	Linux
JDBC driver	Cannot be connected	Cannot be connected
ODBC driver	Cannot be connected	No connection constraints
.NET Data Provider	Can only connect when the encoding system used for definitions is UTF-8	-
SQLEmbedded SQL in C	Can only connect when the connection service file (pg_service.conf) is used	No connection constraints
psql command	Can only connect when the connection service file (pg_service.conf) is used	No connection constraints

A.6 How to Build and Run an Application that Uses Shared Libraries

This section describes the following supplementary items regarding the use of shared libraries when developing applications with Fujitsu Enterprise Postgres.

- Setting DT_RUNPATH for the application
- Direct linking of indirectly used libraries to applications

A.6.1 Setting DT_RUNPATH for Applications

Searching for libraries used by applications

If your application uses Fujitsu Enterprise Postgres shared libraries such as libpq and ecpg, you need to load those libraries when you run your application.

When loading a library, it searches the machine for the library.

Library Search allows you to specify the path to search.

To specify the path, you can use the DT_RUNPATH attribute recorded in the application or library, or the environment variable LD_LIBRARY_PATH.

In general, use of the DT_RUNPATH attribute is recommended. This is because the environment variable LD_LIBRARY_PATH may affect the execution of applications other than the corresponding application.

The following explains when to use DT_RUNPATH.

Build so that the path in the operating environment that stores the library to be used is set in the DT_RUNPATH attribute of the application.

For example, when using libpq or ecpg, set "<*Fujitsu Enterprise Postgres client feature installation directory in operating environment*>/lib". (Note 1)

For how to set the DT_RUNPATH attribute, refer to the documentation of your compiler or linker.

There are the following three types of paths to be set and how to operate them. Please select the one suitable for each operation, and set and operate.

(1) Specify an absolute path

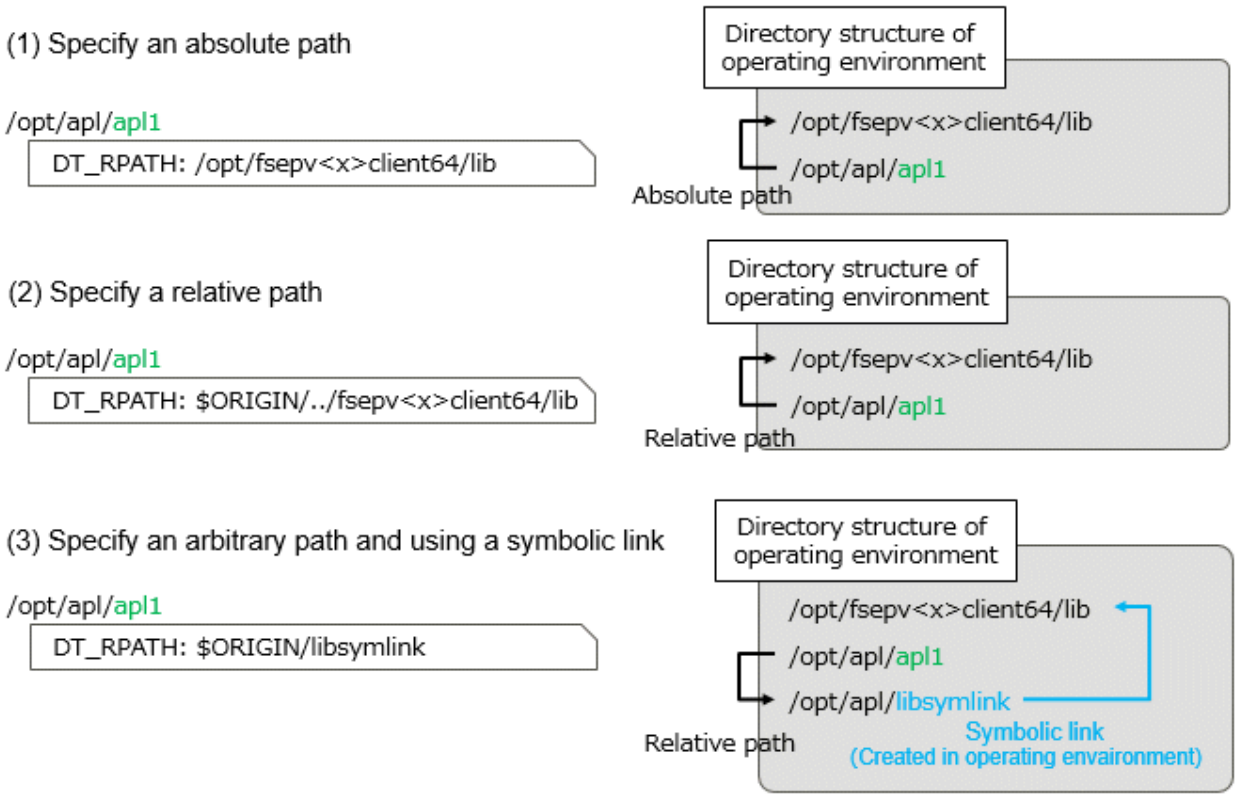
Set "<absolute path of the directory where the library is stored>/lib".

(2) Specify a relative path (Note 1)

Set "<relative path from the application of the directory that stores the library>/lib".

(3) Specify an arbitrary path and using a symbolic link

Set "(any path)". Also, create a symbolic link to the directory that stores the library at any path location.



Note 1:

For example, if you are building an application with gcc, add "-Wl,-rpath,<*Fujitsu Enterprise Postgres installation directory in the operating environment*/lib>,--enable-new-dtags" as gcc options, DT_RUNPATH can be set.

See

For setting DT_RUNPATH, refer to the linker ld's rpath and enable-new-dtags options, for example.

Also, refer to $ORIGIN in ld.so for specifying the relative position from the application.

When DT_RUNPATH cannot be set

If none of the above settings work, you can find the shared libraries your application needs by setting the environment variable LD_LIBRARY_PATH to the path to the directory that stores the libraries when you run your application.

However, if you set LD_LIBRARY_PATH and execute commands or programs other than the applicable application, please be aware that they may result in run-time errors or unexpected behavior.

See

Check the ld.so man page for more information on searching for shared libraries.

A.6.2 Direct Linking of Indirectly Used Libraries to Applications

Note

This content is complicated, and the possibility that it corresponds to this content in many applications is considered to be low.

Use this as a reference only when the application cannot be executed unexpectedly during a test, etc., when creating the application (as explained below, libF3 fails to load, or a runtime error occurs due to conflicts with other libraries).

For applications that use the libraries bundled with Fujitsu Enterprise Postgres, for example, if libF1, libF2, and libF3 have the following dependencies with other applications and libraries as shown below, libF3 can be directly Please specify in the build option to link.

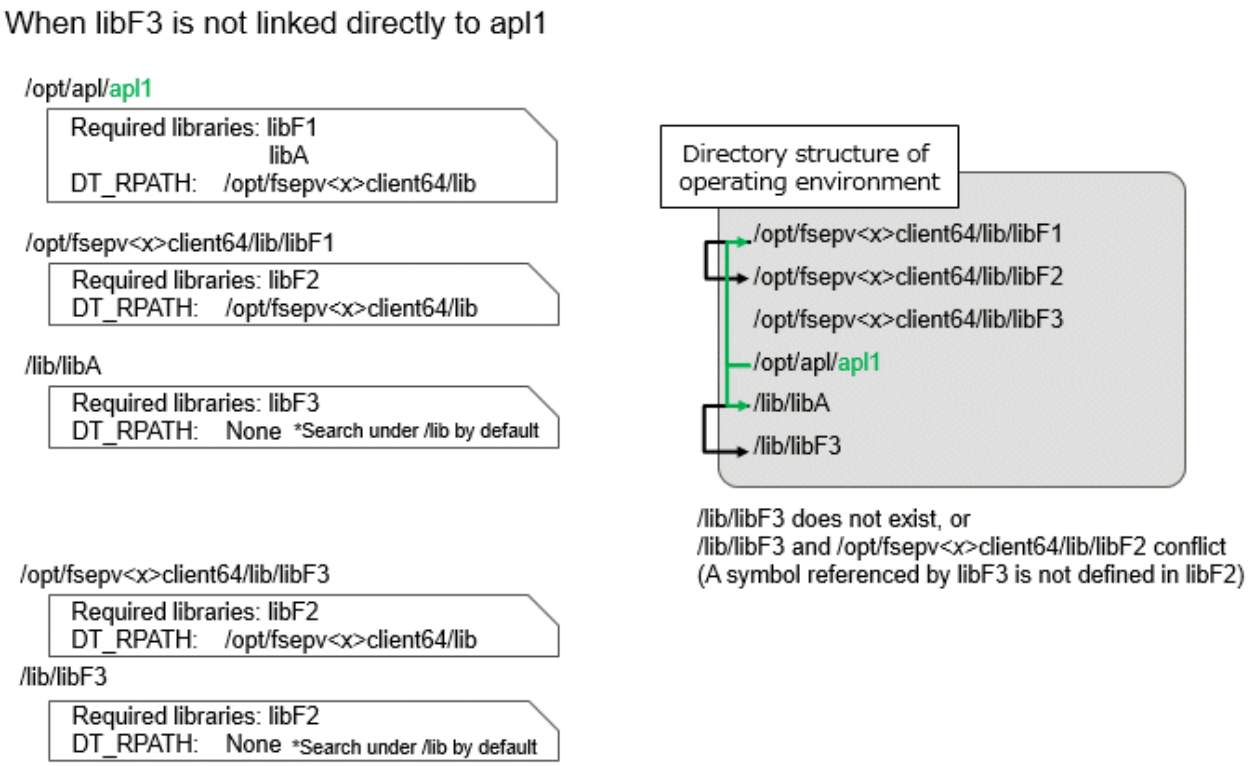
If you do not link directly, specify the path where libF3 is stored in LD_LIBRARY_PATH when running the application.

Otherwise, an error may occur during application execution.

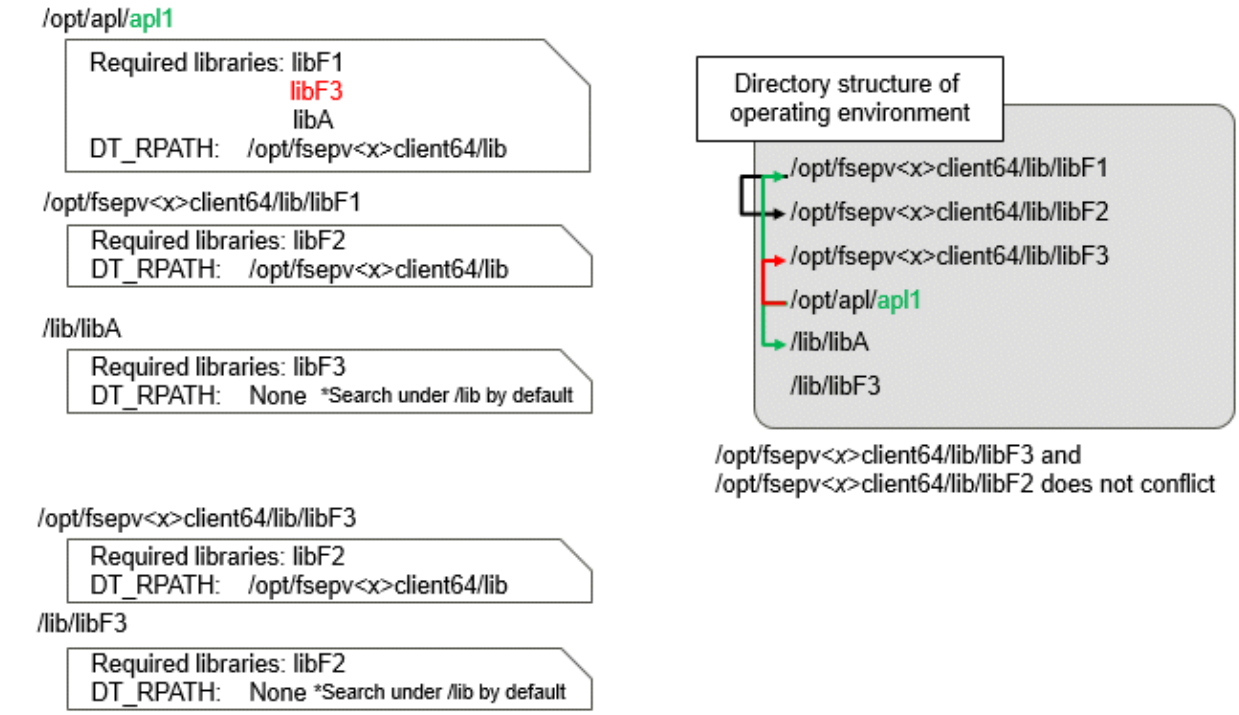
Example

The assumptions for this example are:

- There are the following dependencies between the application and the library.
  - apl1->libF1->libF2 (apl1 requires libF1 and libF1 requires libF2)
  - apl1->libA->libF3->libF2 (apl1 requires libA, libA requires libF3, libF3 requires libF2)
- libF1, libF2, and libF3 are libraries bundled with Fujitsu Enterprise Postgres.



When libF3 is directly linked to apl1



Generally, when loading libraries, if some libA requires libB and libB requires libC, only the DT_RUNPATH value set in libA is used to search libB, and only the DT_RUNPATH value set in libB is used to search libC.

In the figure above "When libF3 is not linked directly to apl1", the search for libF1 uses the DT_RUNPATH value set in apl1. The search for libF2 uses the DT_RUNPATH value set for libF1. Since "*Fujitsu Enterprise Postgres installation directory*/lib" is set in each DT_RUNPATH, libF1 and libF2 bundled with Fujitsu Enterprise Postgres can be found and loaded when searching for libF1 and libF2. .

However, searching for libF3 fails to load libF3 shipped with Fujitsu Enterprise Postgres. This is because the search for libF3 uses the DT_RUNPATH value set for libA, but libA does not have a DT_RUNPATH value.

Therefore, either libF3 cannot be found and loading of libF3 fails, or even if an unexpected libF3 is found in the machine and loaded, there is a conflict between libF2 bundled with Fujitsu Enterprise Postgres conflicts and can result in run-time errors. (The conflict here is that a symbol referenced in libF3 is not defined in libF2.)

By directly linking libF3 to the application, as shown in the figure above "When libF3 is directly linked to apl1", it is possible to use the DT_RUNPATH of the application and load the libF3 bundled with Fujitsu Enterprise Postgres. can. Alternatively, you can load libF3 shipped with Fujitsu Enterprise Postgres by setting LD_LIBRARY_PATH.

Appendix B Conversion Procedures Required due to Differences from Oracle Database

This appendix explains how to convert from an Oracle database to Fujitsu Enterprise Postgres, within the scope noted in "Chapter 9 Compatibility with Oracle Databases" from the following perspectives:

- Feature differences
- Specification differences

This document assumes that the version of the Oracle database to be converted is 7-10.2g.

B.1 Outer Join Operator (Perform Outer Join)

Features

In the WHERE clause conditional expression, by adding the plus sign (+), which is the outer join operator, to the column of the table you want to add as a table join, it is possible to achieve an outer join that is the same as a joined table (OUTER JOIN).

B.1.1 Comparing with the ^= Comparison Operator

Oracle database

```
SELECT *
  FROM t1, t2
  WHERE t1.col1(+) ^= t2.col1;
```

Note: col1 is assumed to be CHAR(4) type

Fujitsu Enterprise Postgres

```
SELECT *
  FROM t1, t2
  WHERE t1.col1(+) != t2.col1;
```

Note: col1 is assumed to be CHAR(4) type

Feature differences

Oracle database

The ^= comparison operator can be specified.

Fujitsu Enterprise Postgres

The ^= comparison operator cannot be specified.

Conversion procedure

Convert using the following procedure:

1. Locate the places where the keyword "^=" is used.
2. Ensure that the keyword, "(+)", is either on the right or left-hand side.
3. Change "^=" to " !=".

B.2 DECODE (Compare Values and Return Corresponding Results)

Features

DECODE compares values of the conversion target value expression and the search values one by one, and if the values of the conversion target value expression and the search values match, a corresponding result value is returned.

B.2.1 Comparing Numeric Data of Character String Types and Numeric Characters

Oracle database

```
SELECT DECODE( col1,
               1000, 'ITEM-A',
               2000, 'ITEM-B',
               'ITEM-C')
  FROM t1;
```

Note: col1 is assumed to be CHAR(4) type

Fujitsu Enterprise Postgres

```
SELECT DECODE( CAST(col1 AS INTEGER),
               1000, 'ITEM-A',
               2000, 'ITEM-B',
               'ITEM-C')
  FROM t1;
```

Note: col1 is assumed to be CHAR(4) type

Feature differences

Oracle database

When the value expression is a string and the search value is a numeric, the string value will be converted to the data type of the comparison target numeric, so that they can be compared.

Fujitsu Enterprise Postgres

If the conversion target value expression is a string value, then no search value can be specified with numbers.

Conversion procedure

Since the data type that can be specified for the conversion target value expression is unknown, use CAST to explicitly convert the conversion target value expression (col1 in the example) to a numeric (INTEGER type in the example).

B.2.2 Obtaining Comparison Result from more than 50 Conditional Expressions

Oracle database

```
SELECT DECODE(col1,
               1,'A',
               2,'B',
               ...
               78,'BZ',
               NULL,'UNKNOWN',
               'OTHER')
   FROM t1;
```

Note: col1 is assumed to be INTEGER type

Fujitsu Enterprise Postgres

```
SELECT CASE
           WHEN col1 = 1 THEN 'A'
           WHEN col1 = 2 THEN 'B'
           ...
           WHEN col1 = 78 THEN 'BZ'
           WHEN col1 IS NULL THEN 'UNKNOWN'
           ELSE 'OTHER'
         END
    FROM t1;
```

Note: col1 is assumed to be INTEGER type

Feature differences

Oracle database

Search value with a maximum of 127 items (up to 255 arguments in total) can be specified.

Fujitsu Enterprise Postgres

Search value with a maximum of 49 items (up to 100 arguments in total) only can be specified.

Conversion procedure

Convert to the CASE expression using the following procedure:

1. Specify the DECODE conversion target value expression (col1 in the first argument, in the example) and the search value (1 in the second argument, in the example) for the CASE expression search condition. Specify the DECODE result value ('A' in the third argument, in the example) for the CASE expression THEN (WHEN col1 = 1 THEN 'A', in the example). Note that if the search value is NULL, specify "IS NULL" for the search condition for the CASE expression.
2. If the DECODE default value ('OTHER' in the last argument, in the example) is specified, specify the default value for the CASE expression ELSE (ELSE 'OTHER', in the example).

B.2.3 Obtaining Comparison Result from Values with Different Data Types

Oracle database

```
SELECT DECODE( col1,
               '1000', 'A',
               '2000', 1,
               'OTHER')
FROM t1;
```

Note: col1 is assumed to be CHAR(4) type

Fujitsu Enterprise Postgres

```
SELECT DECODE( col1,
              '1000', 'A',
              '2000', '1',
              'OTHER')
FROM t1;
```

Note: col1 is assumed to be CHAR(4) type

Feature differences

Oracle database

The data types of all result values are converted to the data type of the first result value.

Fujitsu Enterprise Postgres

Results in an error.

Conversion procedure

Convert using the following procedure:

1. Check the literal data type for the first result value specified.
2. Change the literals specified for each result value to the literal data type checked in the step 1.

B.3 SUBSTR (Extract a String of the Specified Length from Another String)

Features

SUBSTR returns the number of characters specified in the third argument (starting from the position specified in the second argument) from the string specified in the first argument.

Refer to "9.2.2 Notes on SUBSTR" for details on precautions when using SUBSTR.

B.3.1 Specifying a Value Expression with a Data Type Different from the One that can be Specified for Function Arguments

Oracle database

```
SELECT SUBSTR( col1,
               1,
               col2)
  FROM DUAL;
```

Note: col1 and col2 are assumed to be CHAR type

Fujitsu Enterprise Postgres

```
CREATE CAST (CHAR AS INTEGER) WITH INOUT AS IMPLICIT;

SELECT SUBSTR( col1,
               1,
               col2)
  FROM DUAL;
# No changes to SELECT statement;
```

Note: col1 and col2 are assumed to be CHAR type

Feature differences

Oracle database

If the type can be converted to a data type that can be specified for function arguments, conversion is performed implicitly.

Fujitsu Enterprise Postgres

If the data types are different from each other, or if loss of significance occurs, implicit conversion is not performed.

Conversion procedure

Since the data type of the string length is clear, first execute the following CREATE CAST only once so that the CHAR type value (col2 in the example) specified for the string length is implicitly converted to INTEGER type.

```
CREATE CAST (CHAR AS INTEGER) WITH INOUT AS IMPLICIT;
```

B.3.2 Extracting a String with the Specified Format from a Datetime Type Value

Oracle database

```
SELECT SUBSTR( CURRENT_TIMESTAMP,
               1,
               8)
  FROM DUAL;
```

Fujitsu Enterprise Postgres

```
SELECT SUBSTR( TO_CHAR(CURRENT_TIMESTAMP,
               'DD-MON-YY HH.MI.SS.US PM')
                1,
                8)
  FROM DUAL;
```

Feature differences

Oracle database

A datetime value such as CURRENT_TIMESTAMP can be specified for character value expressions.

Fujitsu Enterprise Postgres

A datetime value such as CURRENT_TIMESTAMP cannot be specified for character value expressions.

Conversion procedure

First, specify TO_CHAR for the SUBSTR character value expression.
Specify datetime type (CURRENT_TIMESTAMP, in the example) in firstArg of TO_CHAR, and specify the format template pattern ('DD-MON-YY HH.MI.SS.US PM', in the example) for secondArg to match with the result of SUBSTR before conversion.

TO_CHAR specification format: TO_CHAR(*firstArg, secondArg*)

Information

Refer to "Data Type Formatting Functions" in the PostgreSQL Documentation for information on format template patterns that can be specified for TO_CHAR in Fujitsu Enterprise Postgres.

B.3.3 Concatenating a String Value with a NULL value

Oracle database

```
SELECT SUBSTR( col1 || col2,
               2,
               5)
  FROM t1;
```

Note: col1 and col2 are assumed to be character string type, and col2 may contain NULL

Fujitsu Enterprise Postgres

```
SELECT SUBSTR( col1 || NVL(col2, '')
               2,
               5)
  FROM t1;
```

Note: col1 and col2 are assumed to be character string type, and col2 may contain NULL

Feature differences

Oracle database

NULL is handled as an empty string, and strings are joined.

Fujitsu Enterprise Postgres

NULL is not handled as an empty string, and the result of joining the strings becomes NULL.

Conversion procedure

Convert using the following procedure:

1. Locate the places where the keyword "||" is used.
2. Check if any of the value expressions can contain NULL - if they can, then execute step 3.
3. Modify to NVL(*valExpr,*'').

B.4 NVL (Replace NULL)

Features

NVL converts NULL values.

B.4.1 Obtaining Result from Arguments with Different Data Types

Oracle database

```
SELECT NVL( col1,
            col2)
  FROM t1;
```

Note: col1 is assumed to be VARCHAR(100) type, and col2 is assumed to be CHAR(100) type

Fujitsu Enterprise Postgres

```
SELECT NVL( col1,
            CAST(col2 AS VARCHAR(100)))
  FROM t1;
```

Note: col1 is assumed to be VARCHAR(100) type, and col2 is assumed to be CHAR(100) type

Feature differences

Oracle database

Value expressions with different data types can be specified. If the first argument is a string value, then VARCHAR2 is returned, and if it is a numeric, then a numeric type with greater range is returned.

Fujitsu Enterprise Postgres

Value expressions with different data types cannot be specified.

Conversion procedure

Since the data types that can be specified for the expressions in the two arguments are unknown, use the following steps to convert:

1. Check the data types specified for each of the two expressions.
2. Using the data type that is to be received as a result, explicitly convert the other argument with CAST.

B.4.2 Operating on Datetime/Numeric, Including Adding Number of Days to a Particular Day

Oracle database

```
SELECT NVL( col1 + 10, CURRENT_DATE)
  FROM t1;
```

Note: col1 is assumed to be TIMESTAMP WITHOUT TIME ZONE type or TIMESTAMP WITH TIME ZONE type

Fujitsu Enterprise Postgres

```
SELECT NVL( CAST(col1 AS DATE) + 10, CURRENT_DATE)
  FROM t1;
```

Note: col1 is assumed to be TIMESTAMP WITHOUT TIME ZONE type or TIMESTAMP WITH TIME ZONE type

Feature differences

Oracle database

Numbers can be operated (added to or subtracted from) with either TIMESTAMP WITHOUT TIME ZONE type or TIMESTAMP WITH TIME ZONE type. Operation result will be DATE type.

Fujitsu Enterprise Postgres

Numbers cannot be operated (added to or subtracted from) with neither TIMESTAMP WITHOUT TIME ZONE type nor TIMESTAMP WITH TIME ZONE type. However, numbers can be operated (added to or subtracted from) with DATE type.

Conversion procedure

Convert using the following procedure:

1. Search locations where the keyword "+" or "-" is used in addition or subtraction, and check if these operations are between numbers and TIMESTAMP WITHOUT TIME ZONE type or TIMESTAMP WITH TIME ZONE type.
2. If they are, use CAST to explicitly convert TIMESTAMP WITHOUT TIME ZONE type or TIMESTAMP WITH TIME ZONE type to DATE type.

B.4.3 Calculating INTERVAL Values, Including Adding Periods to a Date

Oracle database

```
SELECT NVL( CURRENT_DATE + (col1 * 1.5), col2)
  FROM t1;
```

Note: col1 and col2 are assumed to be INTERVAL YEAR TO MONTH types

Fujitsu Enterprise Postgres

```
SELECT NVL( CURRENT_DATE +
        CAST(col1 * 1.5 AS
```

```
          INTERVAL YEAR TO MONTH), col2)
  FROM t1;
```

Note: col1 and col2 are assumed to be INTERVAL YEAR TO MONTH types

Feature differences

Oracle database

INTERVAL YEAR TO MONTH type multiplication and division result in INTERVAL YEAR TO MONTH type and any fraction (number of days) will be truncated.

Fujitsu Enterprise Postgres

INTERVAL YEAR TO MONTH type multiplication and division result in INTERVAL type and fractions (number of days) will not be truncated.

Conversion procedure

Convert using the following procedure:

1. Search locations where the keywords "*" or "/" are used in multiplication or division, and check if the specified value is INTERVAL YEAR TO MONTH type.
2. If the value is INTERVAL YEAR TO MONTH type, use CAST to explicitly convert the operation result to INTERVAL YEAR TO MONTH type.

B.5 DBMS_OUTPUT (Output Messages)

Features

DBMS_OUTPUT sends messages to clients such as psql from PL/pgSQL.

B.5.1 Outputting Messages Such As Process Progress Status

Oracle database

```
set serveroutput on;...(1)

DECLARE
  v_col1        CHAR(20);
  v_col2        INTEGER;
  CURSOR c1 IS
    SELECT col1, col2 FROM t1;
BEGIN
  DBMS_OUTPUT.PUT_LINE('-- BATCH_001 Start --');
  OPEN c1;
  DBMS_OUTPUT.PUT_LINE('-- LOOP Start --');
  LOOP
    FETCH c1 INTO v_col1, v_col2;
    EXIT WHEN c1%NOTFOUND;
    DBMS_OUTPUT.PUT('.');
  END LOOP;
  DBMS_OUTPUT.NEW_LINE; ...(2)
  DBMS_OUTPUT.PUT_LINE('-- LOOP End --');
  CLOSE c1;

  DBMS_OUTPUT.PUT_LINE('-- BATCH_001 End --');


EXCEPTION
  WHEN OTHERS THEN
     DBMS_OUTPUT.PUT_LINE('-- SQL Error --');
     DBMS_OUTPUT.PUT_LINE('ERROR : ' || SQLERRM );
```

```
END;
/
```

Fujitsu Enterprise Postgres

```
DO $$
DECLARE
    v_col1      CHAR(20);
    v_col2      INTEGER;
    c1 CURSOR FOR
        SELECT col1, col2 FROM t1;
BEGIN
    PERFORM DBMS_OUTPUT.SERVEROUTPUT(TRUE); ...(1)
    PERFORM DBMS_OUTPUT.ENABLE(NULL); ...(1)

    PERFORM DBMS_OUTPUT.PUT_LINE('-- BATCH_001 Start --');

    OPEN c1;
    PERFORM DBMS_OUTPUT.PUT_LINE('-- LOOP Start --');
    LOOP
        FETCH c1 INTO v_col1, v_col2;
        EXIT WHEN FOUND = false;
        PERFORM DBMS_OUTPUT.PUT('.');
    END LOOP;
    PERFORM DBMS_OUTPUT.NEW_LINE(); ...(2)

    PERFORM DBMS_OUTPUT.PUT_LINE('-- LOOP End --');
    CLOSE c1;

    PERFORM DBMS_OUTPUT.PUT_LINE('-- BATCH_001 End --');

EXCEPTION
    WHEN OTHERS THEN
        PERFORM DBMS_OUTPUT.PUT_LINE('-- SQL Error --');
        PERFORM DBMS_OUTPUT.PUT_LINE('ERROR : ' || SQLERRM );
END;
$$
;
```

(1) SERVEROUTPUT/ENABLE

Specification differences

Oracle database

Use SET statement and specify SERVEROUTPUT ON.

Fujitsu Enterprise Postgres

Specify DBMS_SQL.SERVEROUTPUT(TRUE).

Conversion procedure

Convert using the following procedure:

1. Check if a SET SERVEROUTPUT statement is specified before the PL/SQL block of a stored procedure.
2. If a SET SERVEROUTPUT statement is specified, specify DBMS_SQL.SERVEROUTPUT straight after BEGIN of PL/pgSQL. If ON is specified to have messages output to a window, then specify TRUE. If OFF is specified, then specify FALSE.
3. Specify DBMS_SQL.ENABLE only if SET SERVEROUTPUT is ON. The values to be specified for the argument are as follows:
   - If SIZE is specified for the SET SERVEROUTPUT statement, specify this size for the argument.

- If SIZE is not specified for the SET SERVEROUTPUT statement, then specify 2000 for Oracle10.1g or earlier, NULL for Oracle10.2g or later.

If DBMS_SQL.ENABLE is specified for the PL/SQL block of the stored procedure, specify the same value as that argument.

(2) NEW_LINE

Specification differences

Oracle database

If there is no argument for *packageName.featureName*, parenthesis can be omitted.

Fujitsu Enterprise Postgres

Even if there is no argument for *packageName.featureName*, parenthesis cannot be omitted.

Conversion procedure

Convert using the following procedure:

1. Locate the places where the keyword "DBMS_OUTPUT.NEW_LINE" is used in the stored procedure.
2. If there is no parenthesis after *packageName.featureName*, add the parenthesis.

B.5.2 Receiving a Return Value from a Procedure (PL/SQL) Block (For GET_LINES)

Oracle database

```
set serveroutput off;

DECLARE
    v_num        INTEGER;
BEGIN

    DBMS_OUTPUT.DISABLE; ...(3)
    DBMS_OUTPUT.ENABLE(20000); ...(4)
    DBMS_OUTPUT.PUT_LINE('-- ITEM CHECK --');


    SELECT count(*) INTO v_num FROM t1;

    IF v_num = 0 THEN
        DBMS_OUTPUT.PUT_LINE('-- NO ITEM --');

    ELSE
        DBMS_OUTPUT.PUT_LINE('-- IN ITEM(' || v_num || ') --');
    END IF;
END;
/

set serveroutput on;

DECLARE
    v_buffs      DBMSOUTPUT_LINESARRAY; ...(5)
    v_num        INTEGER := 10;
BEGIN

    DBMS_OUTPUT.GET_LINES(v_buffs, v_num); ...(5)

    FOR i IN 1..v_num LOOP
        DBMS_OUTPUT.PUT_LINE('LOG : ' || v_buffs(i)); ...(5)
```

```
    END LOOP;
END;
/
```

Fujitsu Enterprise Postgres

```
DO $$
DECLARE
    v_num        INTEGER;
BEGIN
    PERFORM DBMS_OUTPUT.SERVEROUTPUT(FALSE);
    PERFORM DBMS_OUTPUT.DISABLE(); ...(3)
    PERFORM DBMS_OUTPUT.ENABLE(20000); ...(4)
PERFORM DBMS_OUTPUT.PUT_LINE('-- ITEM CHECK --');

    SELECT count(*) INTO v_num FROM t1;

    IF v_num = 0 THEN
        PERFORM DBMS_OUTPUT.PUT_LINE('-- NO ITEM --');
    ELSE
        PERFORM DBMS_OUTPUT.PUT_LINE('-- IN ITEM(' || v_num || ') --');
    END IF;
END;
$$
;

DO $$
DECLARE
    v_buffs      VARCHAR[]; ...(5)
    v_num        INTEGER := 10;
BEGIN
    PERFORM DBMS_OUTPUT.SERVEROUTPUT(TRUE);
    SELECT lines, numlines INTO v_buffs, v_num FROM DBMS_OUTPUT.GET_LINES(v_num); ...(5)

    FOR i IN 1..v_num LOOP
        PERFORM DBMS_OUTPUT.PUT_LINE('LOG : ' || v_buffs[i]); ...(5)
    END LOOP;
END;
$$
;
```

(3) DISABLE

Same as the NEW_LINE in the DBMS_OUTPUT package. Refer to NEW_LINE for information on specification differences and conversion procedures associated with specification differences.

(4) ENABLE

Same as NEW_LINE in the DBMS_OUTPUT package. Refer to NEW_LINE for information on specification differences and conversion procedures associated with specification differences.

(5) GET_LINES

Specification format for Oracle database

DBMS_OUTPUT.GET_LINES(*firstArg, secondArg*)

Specification differences

Oracle database

Obtained values are received with variables specified for arguments.

Fujitsu Enterprise Postgres

Since obtained values are the search results for DBMS_OUTPUT.GET_LINES, they are received with variables specified for the INTO clause of the SELECT statement.

Conversion procedure

Convert using the following procedure:

1. Locate the places where the keyword "DBMS_OUTPUT.GET_LINES" is used in the stored procedure.
2. Change the data type (DBMSOUTPUT_LINESARRAY in the example) of the variable (v_buffs in the example) specified as *firstArg* of DBMS_OUTPUT.GET_LINES into a VARCHAR type array (VARCHAR[] in the example).
3. Replace the DBMS_OUTPUT.GET_LINES location called with a SELECT INTO statement.
   - Use the literal "lines, numlines" in the select list.
   - Specify *firstArg* (v_buffs in the example) and *secondArg* (v_num in the example) configured in DBMS_OUTPUT.GET_LINES, in the INTO clause.
   - Use DBMS_OUTPUT.GET_LINES in the FROM clause. Specify only *secondArg* (v_num in the example) before modification.
4. Identify the location that references *firstArg* (v_buffs in the example), and change it to the PL/pgSQL array reference format (v_buffs[i] in the example).

B.5.3 Receiving a Return Value from a Procedure (PL/SQL) Block (For GET_LINE)

Oracle database

```
set serveroutput on;

DECLARE
    v_buff1       VARCHAR2(100);
    v_buff2       VARCHAR2(1000);
    v_num        INTEGER;
BEGIN

    v_buff2 := '';
    LOOP
        DBMS_OUTPUT.GET_LINE(v_buff1, v_num); ...(6)
        EXIT WHEN v_num = 1;
        v_buff2 := v_buff2 || v_buff1;
    END LOOP;

    DBMS_OUTPUT.PUT_LINE(v_buff2);
END;
/
```

Note: Only the process to obtain a value is stated

Fujitsu Enterprise Postgres

```
DO $$
DECLARE
    v_buff1       VARCHAR(100);
    v_buff2       VARCHAR(1000);
    v_num        INTEGER;
BEGIN
    PERFORM DBMS_OUTPUT.SERVEROUTPUT(TRUE);
    v_buff2 := '';
    LOOP
```

```
        SELECT line, status INTO v_buff1, v_num FROM DBMS_OUTPUT.GET_LINE(); ...(6)
        EXIT WHEN v_num = 1;
        v_buff2 := v_buff2 || v_buff1;
    END LOOP;

    PERFORM DBMS_OUTPUT.PUT_LINE(v_buff2);
END;
$$
;
```

Note: Only the process to obtain a value is stated

(6) GET_LINE

Specification format for Oracle database

DBMS_OUTPUT.GET_LINE(*firstArg, secondArg*)

Specification differences

Oracle database

Obtained values are received with variables specified for arguments.

Fujitsu Enterprise Postgres

Since obtained values are the search results for DBMS_OUTPUT.GET_LINES, they are received with variables specified for the INTO clause of the SELECT statement.

Conversion procedure

Convert using the following procedure:

1. Locate the places where the keyword "DBMS_OUTPUT.GET_LINE" is used in the stored procedure.
2. Replace the DBMS_OUTPUT.GET_LINE location called with a SELECT INTO statement.
   - Use the literal "line, status" in the select list.
   - Specify *firstArg* (v_buff1 in the example) and *secondArg* (v_num in the example) configured in DBMS_OUTPUT.GET_LINE, in the INTO clause.
   - Use DBMS_OUTPUT.GET_LINE in the FROM clause. Although arguments are not specified, parenthesis must be specified.

B.6 UTL_FILE (Perform File Operation)

Features

UTL_FILE reads and writes text files from PL/pgSQL.

B.6.1 Registering a Directory to Load and Write Text Files

Oracle database

```
[Oracle9i or earlier]
Configure the following with initialization parameter
   UTL_FILE_DIR='/home/fsep' ...(1)

[Oracle9.2i or later]
   Configure the following with CREATE DIRECTORY statement
   CREATE DIRECTORY DIR AS '/home/fsep'; ...(1)
```

Fujitsu Enterprise Postgres

```
INSERT INTO UTL_FILE.UTL_FILE_DIR(dir)
       VALUES('/home/fsep'); ...(1)
```

(1) UTL_FILE_DIR/CREATE DIRECTORY

Feature differences

Oracle database

Configure the directory to be operated, using the CREATE DIRECTORY statement or the initialization parameter UTL_FILE_DIR.

Fujitsu Enterprise Postgres

The directory to be operated cannot be configured using the CREATE DIRECTORY statement or the initialization parameter UTL_FILE_DIR.

Conversion procedure

Configure the target directory information in the UTL_FILE.UTL_FILE_DIR table using the INSERT statement. Note that this conversion procedure should be performed only once before executing the PL/pgSQL function.

- When using the initialization parameter UTL_FILE_DIR:

  1. Check the initialization parameter UTL_FILE_DIR value ('/home/fsep' in the example).
  2. Using the INSERT statement, specify and execute the directory name checked in step 1.
     - Specify UTL_FILE.UTL_FILE_DIR(dir) for the INTO clause.
     - Using the character string literal ('/home/fsep' in the example), specify the target directory name for the VALUES clause.
     - If multiple directories are specified, execute the INSERT statement for each directory.

- When using the CREATE DIRECTORY statement:

  1. Check the directory name ('/home/fsep' in the example) registered with the CREATE DIRECTORY statement. To check, log in SQL*Plus as a user with DBA privileges, and execute "show ALL_DIRECTORIES;".
  2. Using the INSERT statement, specify and execute the directory name checked in step 1. Same steps are used to specify the INSERT statement as when using the initialization parameter UTL_FILE_DIR.

B.6.2 Checking File Information

Oracle database

```
CREATE PROCEDURE read_file(fname VARCHAR2) AS


    v_file      UTL_FILE.FILE_TYPE;
    v_exists    BOOLEAN;
    v_length    NUMBER;
    v_bsize     INTEGER;
    v_rbuff     VARCHAR2(1024);
BEGIN


    UTL_FILE.FGETATTR('DIR', fname, v_exists, v_length, v_bsize); ...(2)


    IF v_exists <> true THEN
        DBMS_OUTPUT.PUT_LINE('-- FILE NOT FOUND --');
        RETURN;
    END IF;
```

```
    DBMS_OUTPUT.PUT_LINE('-- FILE DATA --');

    v_file := UTL_FILE.FOPEN('DIR', fname, 'r', 1024); ...(3)
    FOR i IN 1..3 LOOP
        UTL_FILE.GET_LINE(v_file, v_rbuff, 1024); ...(4)
        DBMS_OUTPUT.PUT_LINE(v_rbuff);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('... more');
    DBMS_OUTPUT.PUT_LINE('-- READ END --');

    UTL_FILE.FCLOSE(v_file); ...(5)
    RETURN;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('-- FILE END --');

        UTL_FILE.FCLOSE(v_file);
        RETURN;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('-- SQL Error --');

        DBMS_OUTPUT.PUT_LINE('ERROR : ' || SQLERRM );
        UTL_FILE.FCLOSE_ALL; ...(6)
        RETURN;
END;
/

set serveroutput on

call read_file('file01.txt');
```

Fujitsu Enterprise Postgres

```
CREATE FUNCTION read_file(fname VARCHAR) RETURNS void AS $$
DECLARE
    v_file      UTL_FILE.FILE_TYPE;
    v_exists    BOOLEAN;
    v_length    NUMERIC;
    v_bsize     INTEGER;
    v_rbuff     VARCHAR(1024);
BEGIN
    PERFORM DBMS_OUTPUT.SERVEROUTPUT(TRUE);

    SELECT fexists, file_length, blocksize
       INTO v_exists, v_length, v_bsize
       FROM UTL_FILE.FGETATTR('/home/fsep', fname); ...(2)
    IF v_exists <> true THEN
        PERFORM DBMS_OUTPUT.PUT_LINE('-- FILE NOT FOUND --');
        RETURN;
    END IF;

    PERFORM DBMS_OUTPUT.PUT_LINE('-- FILE DATA --');
    v_file := UTL_FILE.FOPEN('/home/fsep', fname, 'w', 1024); ...(3)
    FOR i IN 1..3 LOOP
        v_rbuff := UTL_FILE.GET_LINE(v_file, 1024); ...(4)
        PERFORM DBMS_OUTPUT.PUT_LINE(v_rbuff);
    END LOOP;
    PERFORM DBMS_OUTPUT.PUT_LINE('... more');
    PERFORM DBMS_OUTPUT.PUT_LINE('-- READ END --');

    v_file := UTL_FILE.FCLOSE(v_file); ...(5)
```

```
    RETURN;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        PERFORM DBMS_OUTPUT.PUT_LINE('-- FILE END --');
        v_file := UTL_FILE.FCLOSE(v_file);
        RETURN;
    WHEN OTHERS THEN
        PERFORM DBMS_OUTPUT.PUT_LINE('-- SQL Error --');
        PERFORM DBMS_OUTPUT.PUT_LINE('ERROR : ' || SQLERRM );
        PERFORM UTL_FILE.FCLOSE_ALL(); ...(6)
        RETURN;
END;
$$
LANGUAGE plpgsql;

SELECT read_file('file01.txt');
```

(2) FGETATTR

Specification format for Oracle database

UTL_FILE.FGETATTR(*firstArg, secondArg*, *thirdArg*, *fourthArg*, *fifthArg*)

Feature differences

Oracle database

If using a CREATE DIRECTORY statement (Oracle9.2i or later), specify a directory object name for the directory name.

Fujitsu Enterprise Postgres

A directory object name cannot be specified for the directory name.

Specification differences

Oracle database

Obtained values are received with variables specified for arguments.

Fujitsu Enterprise Postgres

Since obtained values are the search results for UTL_FILE.FGETATTR, they are received with variables specified for the INTO clause of the SELECT statement.

Conversion procedure

Convert using the following procedure. Refer to UTL_FILE_DIR/CREATE DIRECTORY for information on how to check if the directory object name corresponds to the actual directory name.

1. Locate the places where the keyword "UTL_FILE.FOPEN" is used in the stored procedure.
2. Check the actual directory name ('/home/fsep' in the example) that corresponds to the directory object name ('DIR' in the example).
3. Replace the directory object name ('DIR' in the example) in *firstArg* with the actual directory name ('/home/fsep' in the example) verified in step 2.
4. Replace the UTL_FILE.FGETATTR location called with a SELECT INTO statement.
   - Use the literal "fexists, file_length, blocksize" in the select list.
   - Specify *thirdArg*, *fourthArg*, and *fifthArg* (v_exists, v_length, v_bsize, in the example) specified for UTL_FILE.FGETATTR to the INTO clause in the same order as that of the arguments.
   - Use UTL_FILE.FGETATTR in the FROM clause. Specify only the actual directory name for *firstArg* ('/home/fsep' in the example) and *secondArg* (fname in the example) before modification for the arguments.

(3) FOPEN

Specification format for Oracle

UTL_FILE.FOPEN(*firstArg, secondArg*, *thirdArg*, *fourthArg*, *fifthArg*)

Feature differences

Oracle database

If using a CREATE DIRECTORY statement (Oracle9.2i or later), specify a directory object name for the directory name.

Fujitsu Enterprise Postgres

A directory object name cannot be specified for the directory name.

Conversion procedure

Convert using the following procedure. Refer to UTL_FILE_DIR/CREATE DIRECTORY for information on how to check if the directory object name corresponds to the actual directory name.

1. Locate the places where the keyword "UTL_FILE.FOPEN" is used in the stored procedure.
2. Check the actual directory name ('/home/fsep' in the example) that corresponds to the directory object name ('DIR' in the example).
3. Replace the directory object name ('DIR' in the example) in *firstArg* with the actual directory name ('/home/fsep' in the example) checked in step 1.

(4) GET_LINE

Specification format for Oracle database

UTL_FILE.GET_LINE(*firstArg, secondArg*, *thirdArg*, *fourthArg*)

Specification differences

Oracle database

Obtained values are received with variables specified for arguments.

Fujitsu Enterprise Postgres

Since obtained values are the returned value of UTL_FILE.GET_LINE, they are received with variables specified for substitution statement.

Conversion procedure

Convert using the following procedure:

1. Locate the places where the keyword "UTL_FILE.GET_LINE" is used in the stored procedure.
2. Replace the UTL_FILE.GET_LINE location called with a value assignment (:=).
   - On the left-hand side, specify *secondArg* (v_rbuff in the example) specified for UTL_FILE.GET_LINE.
   - Use UTL_FILE.GET_LINE in the right-hand side. Specify only *firstArg* (v_file in the example) and *thirdArg* (1024 in the example) before modification.

(5) FCLOSE

Specification format for Oracle database

UTL_FILE.FCLOSE(*firstArg*)

Specification differences

Oracle database

After closing, the file handler specified for the argument becomes NULL.

Fujitsu Enterprise Postgres

After closing, set the file handler to NULL by assigning the return value of UTL_FILE.FCLOSE to it.

Conversion procedure

Convert using the following procedure:

1. Locate the places where the keyword "UTL_FILE.FCLOSE" is used in the stored procedure.
2. Replace the UTL_FILE.FCLOSE location called with a value assignment (:=) so that the file handler (v_file in the example) becomes NULL.
   - On the left-hand side, specify the argument (v_file in the example) specified for UTL_FILE.FCLOSE.
   - Use UTL_FILE.FCLOSE in the right-hand side. For the argument, specify the same value (v_file in the example) as before modification.

(6) FCLOSE_ALL

Same as NEW_LINE in the DBMS_OUTPUT package. Refer to NEW_LINE in the DBMS_OUTPUT for information on specification differences and conversion procedures associated with specification differences.

B.6.3 Copying Files

Oracle database

```
CREATE PROCEDURE copy_file(fromname VARCHAR2, toname VARCHAR2) AS
BEGIN


    UTL_FILE.FCOPY('DIR1', fromname, 'DIR2', toname, 1, NULL); ...(7)

    RETURN;

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('-- SQL Error --');

        DBMS_OUTPUT.PUT_LINE('ERROR : ' || SQLERRM );
        RETURN;
END;
/

set serveroutput on

call copy_file('file01.txt','file01_bk.txt');
```

Fujitsu Enterprise Postgres

```
CREATE FUNCTION copy_file(fromname VARCHAR, toname VARCHAR) RETURNS void AS $$
BEGIN
    PERFORM DBMS_OUTPUT.SERVEROUTPUT(TRUE);

    PERFORM UTL_FILE.FCOPY('/home/fsep', fromname, '/home/backup', toname, 1, NULL); ...(7)
    RETURN;

EXCEPTION
    WHEN OTHERS THEN
        PERFORM DBMS_OUTPUT.PUT_LINE('-- SQL Error --');
        PERFORM DBMS_OUTPUT.PUT_LINE('ERROR : ' || SQLERRM );
        RETURN;
END;
```

```
$$
LANGUAGE plpgsql;


SELECT copy_file('file01.txt','file01_bk.txt');
```

(7) FCOPY

Specification format for Oracle database

UTL_FILE.FCOPY(*firstArg, secondArg*, *thirdArg*, *fourthArg, fifthArg, sixthArg*)

Feature differences

Oracle database

If using a CREATE DIRECTORY statement (Oracle9.2i or later), specify a directory object name for the directory name.

Fujitsu Enterprise Postgres

A directory object name cannot be specified for the directory name.

Conversion procedure

Convert using the following procedure. Refer to UTL_FILE_DIR/CREATE DIRECTORY for information on how to check if the directory object name corresponds to the actual directory name.

1. Locate the places where the keyword "UTL_FILE.FCOPY" is used in the stored procedure.
2. Check the actual directory names ('/home/fsep' and '/home/backup', in the example) that correspond to the directory object names ('DIR1' and 'DIR2', in the example) of *firstArg* and *thirdArg* argument.
3. Replace the directory object name ('DIR1' and 'DIR2', in the example) with the actual directory names ('/home/fsep' in the example) checked in step 1.

B.6.4 Moving/Renaming Files

Oracle database

```
CREATE PROCEDURE move_file(fromname VARCHAR2, toname VARCHAR2) AS
BEGIN


    UTL_FILE.FRENAME('DIR1', fromname, 'DIR2', toname, FALSE); ...(8)
    RETURN;

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('-- SQL Error --');

        DBMS_OUTPUT.PUT_LINE('ERROR : ' || SQLERRM );
        RETURN;
END;
/

set serveroutput on

call move_file('file01.txt','file02.txt');
```

Fujitsu Enterprise Postgres

```
CREATE FUNCTION move_file(fromname VARCHAR, toname VARCHAR) RETURNS void AS $$
BEGIN
    PERFORM DBMS_OUTPUT.SERVEROUTPUT(TRUE);
```

```
    PERFORM UTL_FILE.FRENAME('/home/fsep', fromname, '/home/backup', toname, FALSE); ...(8)
    RETURN;

EXCEPTION
    WHEN OTHERS THEN
        PERFORM DBMS_OUTPUT.PUT_LINE('-- SQL Error --');
        PERFORM DBMS_OUTPUT.PUT_LINE('ERROR : ' || SQLERRM );
        RETURN;
END;
$$
LANGUAGE plpgsql;


SELECT move_file('file01.txt','file02.txt');
```

(8) FRENAME

Same as FCOPY for the UTL_FILE package. Refer to FCOPY in the UTL_FILE package for information on specification differences and conversion procedures associated with specification differences.

B.7 DBMS_SQL (Execute Dynamic SQL)

Features

For DBMS_SQL, dynamic SQL can be executed from PL/pgSQL.

B.7.1 Searching Using a Cursor

Oracle database

```
CREATE PROCEDURE search_test(h_where CLOB) AS


    str_sql     CLOB;
    v_cnt       INTEGER;
    v_array     DBMS_SQL.VARCHAR2A;
    v_cur       INTEGER;
    v_smpid     INTEGER;
    v_smpnm     VARCHAR2(20);
    v_addbuff   VARCHAR2(20);
    v_smpage    INTEGER;
    errcd       INTEGER;
    length      INTEGER;
    ret         INTEGER;
BEGIN

   str_sql    := 'SELECT smpid, smpnm FROM smp_tbl WHERE ' || h_where || ' ORDER BY smpid';
    v_smpid     := 0;
    v_smpnm     := '';
    v_smpage    := 0;

    v_cur := DBMS_SQL.OPEN_CURSOR; ...(1)

    v_cnt :=
      CEIL(DBMS_LOB.GETLENGTH(str_sql)/1000);
    FOR idx IN 1 .. v_cnt LOOP
        v_array(idx) :=
            DBMS_LOB.SUBSTR(str_sql,
                            1000,
                            (idx-1)*1000+1);
```

```
    END LOOP;
    DBMS_SQL.PARSE(v_cur, v_array, 1, v_cnt, FALSE, DBMS_SQL.NATIVE); ...(2)

    DBMS_SQL.DEFINE_COLUMN(v_cur, 1, v_smpid);

    DBMS_SQL.DEFINE_COLUMN(v_cur, 2, v_smpnm, 10);


    ret := DBMS_SQL.EXECUTE(v_cur);
    LOOP
        v_addbuff := '';

        IF DBMS_SQL.FETCH_ROWS(v_cur) = 0 THEN
            EXIT;
        END IF;

        DBMS_OUTPUT.PUT_LINE('------------------------------------------------------------');
        DBMS_SQL.COLUMN_VALUE(v_cur, 1, v_smpid, errcd, length); ...(3)




        IF errcd = 1405 THEN ...(3)

          DBMS_OUTPUT.PUT_LINE('smpid       = (NULL)');
        ELSE
          DBMS_OUTPUT.PUT_LINE('smpid       = ' || v_smpid);
        END IF;

        DBMS_SQL.COLUMN_VALUE(v_cur, 2, v_smpnm, errcd, length);

        IF errcd = 1406 THEN
          v_addbuff := '... [len=' || length || ']';
        END IF;
        IF errcd = 1405 THEN
          DBMS_OUTPUT.PUT_LINE('v_smpnm     = (NULL)');
        ELSE
          DBMS_OUTPUT.PUT_LINE('v_smpnm     = ' || v_smpnm || v_addbuff );
        END IF;

DBMS_OUTPUT.PUT_LINE('------------------------------------------------------------');

        DBMS_OUTPUT.NEW_LINE;
    END LOOP;

    DBMS_SQL.CLOSE_CURSOR(v_cur); ...(4)

    RETURN;
END;
/

Set serveroutput on

call search_test('smpid < 100');
```

Fujitsu Enterprise Postgres

```
CREATE FUNCTION search_test(h_where text) RETURNS void AS $$
DECLARE
    str_sql     text;


    v_cur       INTEGER;
```

```
    v_smpid     INTEGER;
    v_smpnm     VARCHAR(20);
    v_addbuff   VARCHAR(20);
    v_smpage    INTEGER;
    errcd       INTEGER;
    length      INTEGER;
    ret         INTEGER;
BEGIN
    PERFORM DBMS_OUTPUT.SERVEROUTPUT(TRUE);
   str_sql    := 'SELECT smpid, smpnm FROM smp_tbl WHERE ' || h_where || ' ORDER BY smpid';
    v_smpid     := 0;
    v_smpnm     := '';
    v_smpage    := 0;

    v_cur := DBMS_SQL.OPEN_CURSOR(); ...(1)

    PERFORM DBMS_SQL.PARSE(v_cur, str_sql, 1); ...(2)
    PERFORM DBMS_SQL.DEFINE_COLUMN(v_cur, 1, v_smpid);
    PERFORM DBMS_SQL.DEFINE_COLUMN(v_cur, 2, v_smpnm, 10);

    ret := DBMS_SQL.EXECUTE(v_cur);
    LOOP
        v_addbuff := '';

        IF DBMS_SQL.FETCH_ROWS(v_cur) = 0 THEN
            EXIT;
        END IF;

        PERFORM
DBMS_OUTPUT.PUT_LINE('-----------------------------------------------------------');
        SELECT value,column_error,actual_length
          INTO v_smpid, errcd, length
          FROM DBMS_SQL.COLUMN_VALUE(v_cur,
                                     1,
                                     v_smpid); ...(3)
        IF errcd = 22002 THEN ...(3)
          PERFORM DBMS_OUTPUT.PUT_LINE('smpid       = (NULL)');
        ELSE
          PERFORM DBMS_OUTPUT.PUT_LINE('smpid       = ' || v_smpid);
        END IF;

        SELECT value,column_error,actual_length INTO v_smpnm, errcd, length FROM
DBMS_SQL.COLUMN_VALUE(v_cur, 2, v_smpnm);
        IF errcd = 22001 THEN
          v_addbuff := '... [len=' || length || ']';
        END IF;
        IF errcd = 22002 THEN
          PERFORM DBMS_OUTPUT.PUT_LINE('v_smpnm     = (NULL)');
        ELSE
          PERFORM DBMS_OUTPUT.PUT_LINE('v_smpnm     = ' || v_smpnm || v_addbuff );
        END IF;

        PERFORM
DBMS_OUTPUT.PUT_LINE('-----------------------------------------------------------');
        PERFORM DBMS_OUTPUT.NEW_LINE();
    END LOOP;

    v_cur := DBMS_SQL.CLOSE_CURSOR(v_cur); ...(4)
    RETURN;
END;
$$
LANGUAGE plpgsql;
```

```
SELECT search_test('smpid < 100');
```

(1) OPEN_CURSOR

Same as NEW_LINE in the DBMS_OUTPUT package. Refer to NEW_LINE in the DBMS_OUTPUT package for information on specification differences and conversion procedures associated with specification differences.

(2) PARSE

Specification format for Oracle database

DBMS_SQL.PARSE(*firstArg, secondArg*, *thirdArg*, *fourthArg*, *fifthArg*)

Feature differences

Oracle database

SQL statements can be specified with string table types (VARCHAR2A type, VARCHAR2S type). Specify this for *secondArg*.

DBMS_SQL.NATIVE, DBMS_SQL.V6, DBMS_SQL.V7 can be specified for processing SQL statements.

Fujitsu Enterprise Postgres

SQL statements cannot be specified with string table types.

DBMS_SQL.NATIVE, DBMS_SQL.V6, DBMS_SQL.V7 cannot be specified for processing SQL statements.

Conversion procedure

Convert using the following procedure:

1. Locate the places where the keyword "DBMS_SQL.PARSE" is used in the stored procedure.
2. Check the data type of the SQL statement specified for *secondArg* (v_array in the example).
   - If the data type is either DBMS_SQL.VARCHAR2A type or DBMS_SQL.VARCHAR2S type, then it is a table type specification. Execute step 3 and continue the conversion process.
   - If the data type is neither DBMS_SQL.VARCHAR2A type nor DBMS_SQL.VARCHAR2S type, then it is a string specification. Execute step 7 and continue the conversion process.
3. Check the SQL statement (str_sql in the example) before it was divided into DBMS_SQL.VARCHAR2A type and DBMS_SQL.VARCHAR2S type.
4. Delete the sequence of the processes (processes near FOR idx in the example) where SQL is divided into DBMS_SQL.VARCHAR2A type and DBMS_SQL.VARCHAR2S type.
5. Replace *secondArg* with the SQL statement (str_sql in the example) before it is divided, that was checked in step 2.
6. Delete *thirdArg*, *fourthArg*, and *fifthArg* (v_cnt, FALSE, DBMS_SQL.NATIVE, in the example).
7. If DBMS_SQL.NATIVE, DBMS_SQL.V6, and DBMS_SQL.V7 are specified, then replace *thirdArg* with a numeric literal 1.
   - If either DBMS_SQL.VARCHAR2A type or DBMS_SQL.VARCHAR2S type is used, then *sixthArg* becomes relevant.
   - If neither DBMS_SQL.VARCHAR2A type nor DBMS_SQL.VARCHAR2S type is used, then *thirdArg* becomes relevant.

(3) COLUMN_VALUE

Specification format for Oracle database

DBMS_SQL.COLUMN_VALUE(*firstArg, secondArg*, *thirdArg*, *fourthArg*, *fifthArg*)

Feature differences

Oracle database

The following error codes are returned for column_error.

- 1406: fetched column value was truncated
- 1405: fetched column value is NULL

Fujitsu Enterprise Postgres

The following error codes are returned for column_error.

- 22001: string_data_right_truncation
- 22002: null_value_no_indicator_parameter

Specification differences

Oracle database

Obtained values are received with variables specified for arguments.

Fujitsu Enterprise Postgres

Since obtained values are the search results for DBMS_SQL.COLUMN_VALUE, they are received with variables specified for the INTO clause of the SELECT statement.

Conversion procedure

Convert using the following procedure:

1. Locate the places where the keyword "DBMS_SQL.COLUMN_VALUE" is used in the stored procedure.
2. Replace the DBMS_SQL.COLUMN_VALUE location called with a SELECT INTO statement.
   - Check the number of arguments (v_smpid, errcd, and length in the example) specified after *secondArg* (1 in the example) of DBMS_SQL.COLUMN_VALUE.
   - Specify "value", "column_error", and "actual_length" in the select list, according to the number of arguments checked in the previous step (for example, if only *thirdArg* is specified, then specify "value" only.)
   - Specify *thirdArg*, *fourthArg*, and *fifthArg* (v_smpid, errcd, length in the example) configured for DBMS_SQL.COLUMN_VALUE, for the INTO clause.
   - Use DBMS_SQL.COLUMN_VALUE in the FROM clause. Specify *firstArg*, *secondArg*, and *thirdArg* (v_cur, 1, v_smpid, in the example) before modification.
3. If the *fourthArg* (column_error value in the example) is used, then check the location of the target variable (errcd in the example).
4. If a decision process is performed in the location checked, then modify the values used in the decision process as below:
   - 1406 to 22001
   - 1405 to 22002

(4) CLOSE_CURSOR

Specification format for Oracle database

DBMS_SQL.CLOSE_CURSOR(*firstArg*)

Specification differences

Oracle database

After closing, the cursor specified in *firstArg* becomes NULL.

Fujitsu Enterprise Postgres

After closing, set the cursor to NULL by assigning the return value of DBMS_SQL.CLOSE_CURSOR to it.

Conversion procedure

Convert using the following procedure:

1. Locate the places where the keyword "DBMS_SQL.CLOSE_CURSOR" is used in the stored procedure.
2. Set the cursor to NULL by assigning (:=) the return value of DBMS_SQL.CLOSE_CURSOR to it.
   - On the left-hand side, specify the argument (v_cur in the example) specified for DBMS_SQL.CLOSE_CURSOR.
   - Use DBMS_SQL.CLOSE_CURSOR in the right-hand side. For the argument, specify the same value (v_cur in the example) as before modification.

Appendix C Tables Used by the Features Compatible with Oracle Databases

This chapter describes the tables used by the features compatible with Oracle databases.

C.1 UTL_FILE.UTL_FILE_DIR

Register the directory handled by the UTL_FILE package in the UTL_FILE.UTL_FILE_DIR table.

Name	Type	Description
dir	text	Name of the directory handled by the UTL_FILE package

Appendix D ECOBPG - Embedded SQL in COBOL

This appendix describes application development using embedded SQL in COBOL.

D.1 Precautions when Using Functions and Operators

An embedded SQL program consists of code written in an ordinary programming language, in this case COBOL, mixed with SQL commands in specially marked sections. To build the program, the source code (*.pco) is first passed through the embedded SQL preprocessor, which converts it to an ordinary COBOL program (*.cob), and afterwards it can be processed by a COBOL compiler. (For details about the compiling and linking see "D.9 Processing Embedded SQL Programs".) Converted ECOBPG applications call functions in the libpq library through the embedded SQL library (ecpglib), and communicate with the PostgreSQL server using the normal frontend-backend protocol.

Embedded SQL has advantages over other methods for handling SQL commands from COBOL code. First, it takes care of the tedious passing of information to and from variables in your C program. Second, the SQL code in the program is checked at build time for syntactical correctness. Third, embedded SQL in COBOL is specified in the SQL standard and supported by many other SQL database systems. The PostgreSQL implementation is designed to match this standard as much as possible, and it is usually possible to port embedded SQL programs written for other SQL databases to PostgreSQL with relative ease.

As already stated, programs written for the embedded SQL interface are normal COBOL programs with special code inserted to perform database-related actions. This special code always has the form:

```
EXEC SQL ... END-EXEC
```

These statements syntactically take the place of a COBOL statement. Depending on the particular statement, they can appear at the data division or at the procedure division. Actual executable SQLs need to be placed at the procedure division, and host variable declarations need to be placed at data division. However, the precompiler does not validate their placements. Embedded SQL statements follow the case-sensitivity rules of normal SQL code, and not those of COBOL.

For COBOL code notation, "fixed' or "variable" can be used. In each line, columns 1 to 6 constitute the line number area, and column 7 is the indicator area. Embedded SQL programs also should be placed in area B (column 12 and beyond).

Note that sample code in this document omits indents for each area.

ECOBPG processes or outputs programs according to the COBOL code notation. COBOL code notation is specified using the ecobpg command. Note, however, that the following restrictions apply:

- For "fixed" notation, area B is from columns 12 to 72. Characters in column 73 and beyond are deleted in the precompiled source.
- For "variable" notation, area B is from column 12 to the last column of that record (up to column 251). Characters in column 252 and beyond are deleted in the precompiled source.

ECOBPG accepts as many COBOL statements as possible. Note, however, that the following restrictions apply:

- In declaring host variable section, you can't use debug line.
- Outside of declaring host variable section, you can use debug line, but you can't contain any SQL in debug lines.
- In declaring host variable section, you can't use commas or semicolons as separator. Use space instead.
- EXEC SQL VAR command, it can be used in ECPG, is not available in ECOBPG. Use REDEFINE clause of COBOL instead.

The following sections explain all the embedded SQL statements.

D.2 Managing Database Connections

This section describes how to open, close, and switch database connections.

D.2.1 Connecting to the Database Server

One connects to a database using the following statement:

```
EXEC SQL CONNECT TO target [AS connection-name] [USER user-name] END-EXEC.
```

The target can be specified in the following ways:

- dbname[@hostname][:port]
- tcp:postgresql://hostname[:port][/dbname][?options]
- unix:postgresql://hostname[:port][/dbname][?options]
- an SQL string literal containing one of the above forms
- a reference to a character variable containing one of the above forms (see examples)
- DEFAULT

If you specify the connection target literally (that is, not through a variable reference) and you don't quote the value, then the case-insensitivity rules of normal SQL are applied. In that case you can also double-quote the individual parameters separately as needed. In practice, it is probably less error-prone to use a (single-quoted) string literal or a variable reference. The connection target DEFAULT initiates a connection to the default database under the default user name. No separate user name or connection name can be specified in that case.

There are also different ways to specify the user name:

- username
- username/password
- username IDENTIFIED BY password
- username USING password

As above, the parameters username and password can be an SQL identifier, an SQL string literal, or a reference to a character variable.

The connection-name is used to handle multiple connections in one program. It can be omitted if a program uses only one connection. The most recently opened connection becomes the current connection, which is used by default when an SQL statement is to be executed (see later in this chapter).

Here are some examples of CONNECT statements:

```
EXEC SQL CONNECT TO mydb@sql.mydomain.com END-EXEC.
```

```
EXEC SQL CONNECT TO tcp:postgresql://sql.mydomain.com/mydb AS myconnection USER john END-
EXEC.
```

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 TARGET PIC X(25).
01 USER PIC X(5).
EXEC SQL END DECLARE SECTION END-EXEC.
 ...
MOVE "mydb@sql.mydomain.com" TO TARGET.
MOVE "john" TO USER.
EXEC SQL CONNECT TO :TARGET USER :USER END-EXEC.
```

The last form makes use of the variant referred to above as character variable reference. For this purpose, only fixed-length string(no VARYING) variable can be used. Trailing spaces are ignored. You will see in later sections how COBOL variables can be used in SQL statements when you prefix them with a colon.

Be advised that the format of the connection target is not specified in the SQL standard. So if you want to develop portable applications, you might want to use something based on the last example above to encapsulate the connection target string somewhere.

D.2.2 Choosing a Connection

SQL statements in embedded SQL programs are by default executed on the current connection, that is, the most recently opened one. If an application needs to manage multiple connections, then there are two ways to handle this.

The first option is to explicitly choose a connection for each SQL statement, for example:

```
EXEC SQL AT connection-name SELECT ... END-EXEC.
```

This option is particularly suitable if the application needs to use several connections in mixed order.

The second option is to execute a statement to switch the current connection. That statement is:

```
EXEC SQL SET CONNECTION connection-name END-EXEC.
```

This option is particularly convenient if many statements are to be executed on the same connection.

Here is an example program managing multiple database connections:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
    01 DBNAME PIC X(7).
EXEC SQL END DECLARE SECTION END-EXEC.

    EXEC SQL CONNECT TO testdb1 AS con1 USER testuser END-EXEC.
    EXEC SQL CONNECT TO testdb2 AS con2 USER testuser END-EXEC.
    EXEC SQL CONNECT TO testdb3 AS con3 USER testuser END-EXEC.

*   This query would be executed in the last opened database "testdb3".
    EXEC SQL SELECT current_database() INTO :DBNAME END-EXEC.
    DISPLAY "current=" DBNAME " (should be testdb3)".

*   Using "AT" to run a query in "testdb2"
    EXEC SQL AT con2 SELECT current_database() INTO :DBNAME END-EXEC.
    DISPLAY "current=" DBNAME " (should be testdb2)".

*   Switch the current connection to "testdb1".
    EXEC SQL SET CONNECTION con1 END-EXEC.

    EXEC SQL SELECT current_database() INTO :DBNAME END-EXEC.
    DISPLAY "current=" DBNAME " (should be testdb1)".

    EXEC SQL DISCONNECT ALL END-EXEC.
```

This example would produce this output:

```
current=testdb3 (should be testdb3)
current=testdb2 (should be testdb2)
current=testdb1 (should be testdb1)
```

D.2.3 Closing a Connection

To close a connection, use the following statement:

```
EXEC SQL DISCONNECT [connection] END-EXEC.
```

The connection can be specified in the following ways:

- connection-name
- DEFAULT
- CURRENT
- ALL

If no connection name is specified, the current connection is closed.

It is good style that an application always explicitly disconnect from every connection it opened.

D.3 Running SQL Commands

Any SQL command can be run from within an embedded SQL application. Below are some examples of how to do that.

D.3.1 Executing SQL Statements

Creating a table:

```
EXEC SQL CREATE TABLE foo (number integer, ascii char(16)) END-EXEC.
EXEC SQL CREATE UNIQUE INDEX num1 ON foo(number) END-EXEC.
EXEC SQL COMMIT END-EXEC.
```

Inserting rows:

```
EXEC SQL INSERT INTO foo (number, ascii) VALUES (9999, 'doodad') END-EXEC.
EXEC SQL COMMIT END-EXEC.
```

Deleting rows:

```
EXEC SQL DELETE FROM foo WHERE number = 9999 END-EXEC.
EXEC SQL COMMIT END-EXEC.
```

Updates:

```
EXEC SQL UPDATE foo
    SET ascii = 'foobar'
    WHERE number = 9999 END-EXEC.
EXEC SQL COMMIT END-EXEC.
```

SELECT statements that return a single result row can also be executed using EXEC SQL directly. To handle result sets with multiple rows, an application has to use a cursor; see "D.3.2 Using Cursors" below. (As a special case, an application can fetch multiple rows at once into an array host variable; see "Arrays".)

Single-row select:

```
EXEC SQL SELECT foo INTO :FooBar FROM table1 WHERE ascii = 'doodad' END-EXEC.
```

Also, a configuration parameter can be retrieved with the SHOW command:

```
EXEC SQL SHOW search_path INTO :var END-EXEC.
```

The tokens of the form :something are *host variables*, that is, they refer to variables in the COBOL program. They are explained in "D.4 Using Host Variables".

D.3.2 Using Cursors

To retrieve a result set holding multiple rows, an application has to declare a cursor and fetch each row from the cursor. The steps to use a cursor are the following: declare a cursor, open it, fetch a row from the cursor, repeat, and finally close it.

Select using cursors:

```
EXEC SQL DECLARE foo_bar CURSOR FOR
    SELECT number, ascii FROM foo
    ORDER BY ascii END-EXEC.
EXEC SQL OPEN foo_bar END-EXEC.
EXEC SQL FETCH foo_bar INTO :FooBar, :DooDad END-EXEC.
...
EXEC SQL CLOSE foo_bar END-EXEC.
EXEC SQL COMMIT END-EXEC.
```

For more details about declaration of the cursor, see "D.11.4 DECLARE", and refer to "SQL Commands" in "Reference" in the PostgreSQL Documentation for information on FETCH command.

Note: The ECOBPG DECLARE command does not actually cause a statement to be sent to the PostgreSQL backend. The cursor is opened in the backend (using the backend's DECLARE command) at the point when the OPEN command is executed.

D.3.3 Managing Transactions

In the default mode, statements are committed only when EXEC SQL COMMIT is issued. The embedded SQL interface also supports autocommit of transactions (similar to libpq behavior) via the -t command-line option to ecobpg or via the EXEC SQL SET AUTOCOMMIT TO ON statement. In autocommit mode, each command is automatically committed unless it is inside an explicit transaction block. This mode can be explicitly turned off using EXEC SQL SET AUTOCOMMIT TO OFF.

See

Refer to "ecpg" in "PostgreSQL Client Applications" in the PostgreSQL Documentation for information on -t command-line option to ecobpg.

The following transaction management commands are available:

EXEC SQL COMMIT END-EXEC

Commit an in-progress transaction.

EXEC SQL ROLLBACK END-EXEC

Roll back an in-progress transaction.

EXEC SQL SET AUTOCOMMIT TO ON END-EXEC

Enable autocommit mode.

EXEC SQL SET AUTOCOMMIT TO OFF END-EXEC

Disable autocommit mode. This is the default.

D.3.4 Prepared Statements

When the values to be passed to an SQL statement are not known at compile time, or the same statement is going to be used many times, then prepared statements can be useful.

The statement is prepared using the command PREPARE. For the values that are not known yet, use the placeholder "?":

```
EXEC SQL PREPARE stmt1 FROM "SELECT oid, datname FROM pg_database WHERE oid = ?" END-EXEC.
```

If a statement returns a single row, the application can call EXECUTE after PREPARE to execute the statement, supplying the actual values for the placeholders with a USING clause:

```
EXEC SQL EXECUTE stmt1 INTO :dboid, :dbname USING 1 END-EXEC.
```

If a statement returns multiple rows, the application can use a cursor declared based on the prepared statement. To bind input parameters, the cursor must be opened with a USING clause:

```
EXEC SQL PREPARE stmt1 FROM "SELECT oid,datname FROM pg_database WHERE oid > ?" END-EXEC.
EXEC SQL DECLARE foo_bar CURSOR FOR stmt1 END-EXEC.

* when end of result set reached, break out of while loop
EXEC SQL WHENEVER NOT FOUND GOTO FETCH-END END-EXEC.

EXEC SQL OPEN foo_bar USING 100 END-EXEC.
...
PERFORM NO LIMIT
    EXEC SQL FETCH NEXT FROM foo_bar INTO :dboid, :dbname END-EXEC
END-PERFORM.

FETCH-END.
EXEC SQL CLOSE foo_bar END-EXEC.
```

When you don't need the prepared statement anymore, you should deallocate it:

```
EXEC SQL DEALLOCATE PREPARE name END-EXEC.
```

For more details about PREPARE, see "D.11.10 PREPARE". Also see "D.5 Dynamic SQL" for more details about using placeholders and input parameters.

D.4 Using Host Variables

In "D.3 Running SQL Commands" you saw how you can execute SQL statements from an embedded SQL program. Some of those statements only used fixed values and did not provide a way to insert user-supplied values into statements or have the program process the values returned by the query. Those kinds of statements are not really useful in real applications. This section explains in detail how you can pass data between your COBOL program and the embedded SQL statements using a simple mechanism called host variables. In an embedded SQL program we consider the SQL statements to be guests in the COBOL program code which is the host language. Therefore the variables of the COBOL program are called host variables.

Another way to exchange values between PostgreSQL backends and ECOBPG applications is the use of SQL descriptors, described in "D.6 Using Descriptor Areas".

D.4.1 Overview

Passing data between the COBOL program and the SQL statements is particularly simple in embedded SQL. Instead of having the program paste the data into the statement, which entails various complications, such as properly quoting the value, you can simply write the name of a COBOL variable into the SQL statement, prefixed by a colon. For example:

```
EXEC SQL INSERT INTO sometable VALUES (:v1, 'foo', :v2) END-EXEC.
```

This statements refers to two COBOL variables named v1 and v2 and also uses a regular SQL string literal, to illustrate that you are not restricted to use one kind of data or the other.

This style of inserting COBOL variables in SQL statements works anywhere a value expression is expected in an SQL statement.

D.4.2 Declare Sections

To pass data from the program to the database, for example as parameters in a query, or to pass data from the database back to the program, the COBOL variables that are intended to contain this data need to be declared in specially marked sections, so the embedded SQL preprocessor is made aware of them.

This section starts with:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
```

and ends with:

```
EXEC SQL END DECLARE SECTION END-EXEC.
```

Between those lines, there must be normal COBOL variable declarations, such as:

```
01 INTX PIC S9(9) COMP VALUE 4.
01 FOO PIC X(15).
01 BAR PIC X(15).
```

As you can see, you can optionally assign an initial value to the variable. The variable's scope is determined by the location of its declaring section within the program.

You can have as many declare sections in a program as you like.

The declarations are also echoed to the output file as normal COBOL variables, so there's no need to declare them again. Variables that are not intended to be used in SQL commands can be declared normally outside these special sections.

The definition of a group item also must be listed inside a DECLARE section. Otherwise the preprocessor cannot handle these types since it does not know the definition.

D.4.3 Retrieving Query Results

Now you should be able to pass data generated by your program into an SQL command. But how do you retrieve the results of a query? For that purpose, embedded SQL provides special variants of the usual commands SELECT and FETCH. These commands have a special INTO clause that specifies which host variables the retrieved values are to be stored in. SELECT is used for a query that returns only single row, and FETCH is used for a query that returns multiple rows, using a cursor.

Here is an example:

```
*
* assume this table:
* CREATE TABLE test (a int, b varchar(50));
*

 EXEC SQL BEGIN DECLARE SECTION END-EXEC.
 01 V1 PIC S9(9).
 01 V2 PIC X(50) VARYING.
 EXEC SQL END DECLARE SECTION END-EXEC.

 ...

 EXEC SQL SELECT a, b INTO :V1, :V2 FROM test END-EXEC.
```

So the INTO clause appears between the select list and the FROM clause. The number of elements in the select list and the list after INTO (also called the target list) must be equal.

Here is an example using the command FETCH:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 V1 PIC S9(9).
01 V2 PIC X(50) VARYING.
EXEC SQL END DECLARE SECTION END-EXEC.

 ...

EXEC SQL DECLARE foo CURSOR FOR SELECT a, b FROM test END-EXEC.

 ...

PERFORM WITH
    ...
    EXEC SQL FETCH NEXT FROM foo INTO :V1, :V2 END-EXEC
```

```
    ...
END-PERFORM.
```

Here the INTO clause appears after all the normal clauses.

D.4.4 Type Mapping

When ECOBPG applications exchange values between the PostgreSQL server and the COBOL application, such as when retrieving query results from the server or executing SQL statements with input parameters, the values need to be converted between PostgreSQL data types and host language variable types (specifically COBOL language data types). One of the main points of ECOBPG is that it takes care of this automatically in most cases.

In this respect, there are two kinds of data types: Some simple PostgreSQL data types, such as integer and text, can be read and written by the application directly. Other PostgreSQL data types, such as timestamp and date can only be accessed through character strings. special library functions does not exist in ecobpg. (pgtypes, exists in ECPG, for COBOL is not implemented yet)

"Table D.1 Mapping Between PostgreSQL Data Types and COBOL Variable Types" shows which PostgreSQL data types correspond to which COBOL data types. When you wish to send or receive a value of a given PostgreSQL data type, you should declare a COBOL variable of the corresponding COBOL data type in the declare section.

Table D.1 Mapping Between PostgreSQL Data Types and COBOL Variable Types

PostgreSQL data type	COBOL Host variable type
smallint	PIC S9([1-4]) {BINARY|COMP|COMP-5}
integer	PIC S9([5-9]) {BINARY|COMP|COMP-5}
bigint	PIC S9([10-18]) {BINARY|COMP|COMP-5}
decimal	PIC S9(m)V9(n) PACKED-DECIMAL PIC 9(m)V9(n) DISPLAY (*1) PIC S9(m)V9(n) DISPLAY PIC S9(m)V9(n) DISPLAY SIGN TRAILING [SEPARATE] PIC S9(m)V9(n) DISPLAY SIGN LEADING [SEPARATE]
numeric	(same with decimal)
real	COMP-1
double precision	COMP-2
small serial	PIC S9([1-4]) {BINARY|COMP|COMP-5}
serial	PIC S9([1-9]) {BINARY|COMP|COMP-5}
bigserial	PIC S9([10-18]) {BINARY|COMP|COMP-5}
oid	PIC 9(9) {BINARY|COMP|COMP-5}
character(*n*), varchar(*n*), text	PIC X(*n*), PIC X(*n*) VARYING
name	PIC X(NAMEDATALEN)
boolean	BOOL(*2)
bytea	BYTEA(n)
other types(e.g. timestamp)	PIC X(*n*), PIC X(*n*) VARYING

*1: If no USAGE is specified, host variable is regarded as DISPLAY.
*2: Type definition is added automatically on pre-compiling.
Body of BOOL is PIC X(1). '1' for true and '0' for false.

You can use some pattern of digits for integer(see table), but if database sends big number with more digits than specified, behavior is undefined.

PostgreSQL data type	COBOL Host variable type
VALUE clause can't be used with VARYING. (Can be used with other types) REDEFINE clause can be used, but it won't be validated on pre-compilation (The COBOL compiler will do this).	

Handling Character Strings

To handle SQL character string data types, such as varchar and text, there is a possible way to declare the host variables.

The way is using the PIC X(*n*) VARYING type (we call it VARCHAR type from now on), which is a special type provided by ECOBPG. The definition on type VARCHAR is converted into a group item consists of named variables. A declaration like:

```
01 VAR PIC X(180) VARYING.
```

is converted into:

```
01 VAR.
49 LEN PIC S9(4) COMP-5.
49 ARR PIC X(180).
```

if --varchar-with-named-member option is used, it is converted into:

```
01 VAR.
49 VAR-LEN PIC S9(4) COMP-5.
49 VAR-ARR PIC X(180).
```

You can use level 1 to 48 for VARCHAR. Don't use level 49 variable right after VARCHAR variable. To use a VARCHAR host variable as an input for SQL statement, LEN must be set the length of the string included in ARR.

To use a VARCHAR host variable as an output of SQL statement, the variable must be declared in a sufficient length. If the length is insufficient, it can cause a buffer overrun.

PIC X(*n*) and VARCHAR host variables can also hold values of other SQL types, which will be stored in their string forms.

Accessing Special Data Types

ECOBPG doesn't have special support for date, timestamp, and interval types.
(ECPG has pgtypes, but ECOBPG doesn't.)
You can use PIC X(*n*) or VARCHAR for DB I/O with these types. See "Data Types" section in PostgreSQL's document.

bytea

Handling of bytea types is similar to VARCHAR. The definition of an array of type bytea is converted into a group item consists of named variables.

A declaration like:

```
01 var bytea(100).
```

is converted into:

```
01 var .
49 LEN PIC S9(9) COMP-5 .
49 ARR PIC X(100) .
```

if bytea-with-named-member option is used, it is converted into:

```
01 var .
49 var-LEN PIC S9(9) COMP-5 .
49 var-ARR PIC X(100) .
```

The data item ARR holds binary format data. Unlike VARCHAR, it is not affected by the locale or character encoding when processing data.

Other usage and prohibitions are the same as VARCHAR.

Note

The bytea variable can only be used if byta_output is set to hex.

Host Variables with Nonprimitive Types

As a host variable you can also use arrays, typedefs, and group items.

Arrays

To create and use array variables, OCCURENCE syntax is provided by COBOL.

The typical use case is to retrieve multiple rows from a query result without using a cursor. Without an array, to process a query result consisting of multiple rows, it is required to use a cursor and the FETCH command. But with array host variables, multiple rows can be received at once. The length of the array has to be defined to be able to accommodate all rows, otherwise a buffer overrun will likely occur.

Following example scans the pg_database system table and shows all OIDs and names of the available databases:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 GROUP-ITEM.
    05 DBID PIC S9(9) COMP OCCURS 8.
    05 DBNAME PIC X(16) OCCURS 8.
01 I PIC S9(9) COMP.
EXEC SQL END DECLARE SECTION END-EXEC.


    EXEC SQL CONNECT TO testdb END-EXEC.

*   Retrieve multiple rows into arrays at once.
    EXEC SQL SELECT oid,datname INTO :DBID, :DBNAME FROM pg_database END-EXEC.

    PERFORM VARYING I FROM 1 BY 1 UNTIL I > 8
        DISPLAY "oid=" DBID(I) ", dbname=" DBNAME(I)
    END-PERFORM.

    EXEC SQL COMMIT END-EXEC.
    EXEC SQL DISCONNECT ALL END-EXEC.
```

You can use member of array as simple host variable by specifying subscript of array. For specifying subscript, use C-style "[1]", not COBOL-style "(1)". But subscript starts with 1, according to COBOL syntax.

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 GROUP-ITEM.
    05 DBID PIC S9(9) COMP OCCURS 8.
EXEC SQL END DECLARE SECTION END-EXEC.

    EXEC SQL CONNECT TO testdb END-EXEC.

    EXEC SQL SELECT oid INTO :DBID[1] FROM pg_database WHERE oid=1 END-EXEC.

        DISPLAY "oid=" DBID(1)

    EXEC SQL COMMIT END-EXEC.
    EXEC SQL DISCONNECT ALL END-EXEC.
```

Group Item

A group item whose subordinate item names match the column names of a query result, can be used to retrieve multiple columns at once. The group item enables handling multiple column values in a single host variable.

The following example retrieves OIDs, names, and sizes of the available databases from the pg_database system table by using the pg_database_size() function. In this example, a group item variable dbinfo_t with members whose names match each column in the SELECT result is used to retrieve one result row without putting multiple host variables in the FETCH statement.

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
    01 DBINFO-T TYPEDEF.
       02 OID PIC S9(9) COMP.
       02 DATNAME PIC X(65).
       02 DBSIZE PIC S9(18) COMP.

    01 DBVAL TYPE DBINFO-T.
EXEC SQL END DECLARE SECTION END-EXEC.


    EXEC SQL DECLARE cur1 CURSOR FOR SELECT oid, datname, pg_database_size(oid) AS size
FROM pg_database END-EXEC.
    EXEC SQL OPEN cur1 END-EXEC.

*   when end of result set reached, break out of loop
    EXEC SQL WHENEVER NOT FOUND GOTO END-FETCH END-EXEC.

    PERFORM NO LIMIT
*        Fetch multiple columns into one structure.
        EXEC SQL FETCH FROM cur1 INTO :DBVAL END-EXEC

*        Print members of the structure.
        DISPLAY "oid=" OID ", datname=" DATNAME ", size=" DBSIZE
    END-PERFORM.

    END-FETCH.
    EXEC SQL CLOSE cur1 END-EXEC.
```

group item host variables "absorb" as many columns as the group item as subordinate items. Additional columns can be assigned to other host variables. For example, the above program could also be restructured like this, with the size variable outside the group item:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
    01 DBINFO-T TYPEDEF.
       02 OID PIC S9(9) COMP.
       02 DATNAME PIC X(65).

    01 DBVAL TYPE DBINFO-T.
    01 DBSIZE PIC S9(18) COMP.
EXEC SQL END DECLARE SECTION END-EXEC.


    EXEC SQL DECLARE cur1 CURSOR FOR SELECT oid, datname, pg_database_size(oid) AS size
FROM pg_database END-EXEC.
    EXEC SQL OPEN cur1 END-EXEC.

*   when end of result set reached, break out of loop
    EXEC SQL WHENEVER NOT FOUND GOTO END-FETCH END-EXEC.

    PERFORM NO LIMIT
*       Fetch multiple columns into one structure.
        EXEC SQL FETCH FROM cur1 INTO :DBVAL, :DBSIZE END-EXEC

*       Print members of the structure.
        DISPLAY "oid=" OID ", datname=" DATNAME ", size=" DBSIZE
    END-PERFORM
```

```
    FETCH-END.
    EXEC SQL CLOSE cur1 END-EXEC.
```

You can use only non-nested group items for host variable of SQL statement. Declaration of nested group items are OK, but you must specify non-nested part of group items for SQL. (VARCHAR, is translated to group item on pre-compilation, is not considered as offense of this rule.) When using inner item of group item in SQL, use C-struct like period separated syntax(not COBOL's A OF B). Here is example.

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 NESTED-GROUP.
  02 CHILD1.
    03 A PIC X(10).
    03 B PIC S9(9) COMP.
  02 CHILD2.
    03 A PIC X(10).
    03 B PIC S9(9) COMP.
EXEC SQL END DECLARE SECTION END-EXEC.

* This SQL is valid. CHILD1 has no nested group items.
EXEC SQL SELECT * INTO :NESTED-GROUP.CHILD1 FROM TABLE1 END-EXEC.
```

For specifying basic item of group items, full specification is not needed if the specification is enough for identifying the item. This is from COBOL syntax. For more detail, see resources of COBOL syntax.

TYPEDEF

Use the typedef keyword to map new types to already existing types.

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
    01 MYCHARTYPE TYPEDEF PIC X(40).
    01 SERIAL-T TYPEDEF PIC S9(9) COMP.
EXEC SQL END DECLARE SECTION END-EXEC.
```

Note that you could also use:

```
EXEC SQL TYPE SERIAL-T IS PIC S9(9) COMP-5. END-EXEC.
```

This declaration does not need to be part of a declare section.

D.4.5 Handling Nonprimitive SQL Data Types

This section contains information on how to handle nonscalar and user-defined SQL-level data types in ECOBPG applications. Note that this is distinct from the handling of host variables of nonprimitive types, described in the previous section.

Arrays

SQL-level arrays are not directly supported in ECOBPG. It is not possible to simply map an SQL array into a COBOL array host variable. This will result in undefined behavior. Some workarounds exist, however.

If a query accesses elements of an array separately, then this avoids the use of arrays in ECOBPG. Then, a host variable with a type that can be mapped to the element type should be used. For example, if a column type is array of integer, a host variable of type PIC S9(9) COMP can be used. Also if the element type is varchar or text, a host variable of type VARCHAR can be used.

Here is an example. Assume the following table:

```
CREATE TABLE t3 (
    ii integer[]
);

testdb=> SELECT * FROM t3;
     ii
-------------
 {1,2,3,4,5}
(1 row)
```

The following example program retrieves the 4th element of the array and stores it into a host variable of type PIC S9(9) COMP-5:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 II PIC S9(9) COMP.
EXEC SQL END DECLARE SECTION END-EXEC.


EXEC SQL DECLARE cur1 CURSOR FOR SELECT ii[4] FROM t3 END-EXEC.
EXEC SQL OPEN cur1 END-EXEC.


EXEC SQL WHENEVER NOT FOUND GOTO END-FETCH END-EXEC.


PERFORM NO LIMIT
    EXEC SQL FETCH FROM cur1 INTO :II  END-EXEC
    DISPLAY "ii=" II
END-PERFORM.

END-FETCH.
EXEC SQL CLOSE cur1 END-EXEC.
```

To map multiple array elements to the multiple elements in an array type host variables each element of array column and each element of the host variable array have to be managed separately, for example:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 GROUP-ITEM.
    05 II_A PIC S9(9) COMP OCCURS 8.
EXEC SQL END DECLARE SECTION END-EXEC.

EXEC SQL DECLARE cur1 CURSOR FOR SELECT ii[1], ii[2], ii[3], ii[4] FROM t3 END-EXEC.
EXEC SQL OPEN cur1 END-EXEC.

EXEC SQL WHENEVER NOT FOUND GOTO END-FETCH END-EXEC.

PERFORM NO LIMIT
    EXEC SQL FETCH FROM cur1 INTO :II_A[1], :II_A[2], :II_A[3], :II_A[4] END-EXEC
    ...
END-PERFORM.
```

Note again that.

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 GROUP-ITEM.
    05 II_A PIC S9(9) COMP OCCURS 8.
EXEC SQL END DECLARE SECTION END-EXEC.

EXEC SQL DECLARE cur1 CURSOR FOR SELECT ii FROM t3 END-EXEC.
EXEC SQL OPEN cur1 END-EXEC.
```

```
EXEC SQL WHENEVER NOT FOUND GOTO END-FETCH END-EXEC.

PERFORM NO LIMIT
*    WRONG
     EXEC SQL FETCH FROM cur1 INTO :II_A END-EXEC
     ...
END-PERFORM.
```

would not work correctly in this case, because you cannot map an array type column to an array host variable directly.

Another workaround is to store arrays in their external string representation in host variables of type VARCHAR. For more details about this representation.

See

Refer to "Arrays" in "Tutorial" in the PostgreSQL Documentation for information more details about this representation.

Note that this means that the array cannot be accessed naturally as an array in the host program (without further processing that parses the text representation).

Composite Types

Composite types are not directly supported in ECOBPG, but an easy workaround is possible. The available workarounds are similar to the ones described for arrays above: Either access each attribute separately or use the external string representation.

For the following examples, assume the following type and table:

```
CREATE TYPE comp_t AS (intval integer, textval varchar(32));
CREATE TABLE t4 (compval comp_t);
INSERT INTO t4 VALUES ( (256, 'PostgreSQL') );
```

The most obvious solution is to access each attribute separately. The following program retrieves data from the example table by selecting each attribute of the type comp_t separately:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 INTVAL PIC S9(9) COMP.
01 TEXTVAL PIC X(33) VARYING.
EXEC SQL END DECLARE SECTION END-EXEC.

* Put each element of the composite type column in the SELECT list.
EXEC SQL DECLARE cur1 CURSOR FOR SELECT (compval).intval, (compval).textval FROM t4 END-
EXEC.
EXEC SQL OPEN cur1 END-EXEC.

EXEC SQL WHENEVER NOT FOUND GOTO END-FETCH END-EXEC.

PERFORM NO LIMIT
*    Fetch each element of the composite type column into host variables.
     EXEC SQL FETCH FROM cur1 INTO :INTVAL, :TEXTVAL END-EXEC

     DISPLAY "intval=" INTVAL ", textval=" ARR OF TEXTVAL
END-PERFORM.

END-FETCH.
EXEC SQL CLOSE cur1 END-EXEC.
```

To enhance this example, the host variables to store values in the FETCH command can be gathered into one group item. For more details about the host variable in the group item form, see "Group Item". To switch to the group item, the example can be modified as below. The two host variables, intval and textval, become subordinate items of the comp_t group item, and the group item is specified on the FETCH command.

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 COMP-T TYPEDEF.
    02 INTVAL PIC S9(9) COMP.
    02 TEXTVAL PIC X(33) VARYING.

01 COMPVAL TYPE COMP-T.
EXEC SQL END DECLARE SECTION END-EXEC.

* Put each element of the composite type column in the SELECT list.
EXEC SQL DECLARE cur1 CURSOR FOR SELECT (compval).intval, (compval).textval FROM t4 END-
EXEC.
EXEC SQL OPEN cur1 END-EXEC.
EXEC SQL WHENEVER NOT FOUND GOTO END-FETCH END-EXEC.

PERFORM NO LIMIT
*   Put all values in the SELECT list into one structure.
    EXEC SQL FETCH FROM cur1 INTO :COMPVAL END-EXEC

    DISPLAY "intval=" INTVAL ", textval=" ARR OF TEXTVAL
END-PERFORM.

END-FETCH.
EXEC SQL CLOSE cur1 END-EXEC.
```

Although a group item is used in the FETCH command, the attribute names in the SELECT clause are specified one by one. This can be enhanced by using a * to ask for all attributes of the composite type value.

```
...
EXEC SQL DECLARE cur1 CURSOR FOR SELECT (compval).* FROM t4 END-EXEC.
EXEC SQL OPEN cur1 END-EXEC.
EXEC SQL WHENEVER NOT FOUND GOTO END-FETCH END-EXEC.

PERFORM NO LIMIT
*   Put all values in the SELECT list into one structure.
    EXEC SQL FETCH FROM cur1 INTO :COMPVAL END-EXEC

    DISPLAY "intval=" INTVAL ", textval=" ARR OF TEXTVAL
END-PERFORM.
```

This way, composite types can be mapped into structures almost seamlessly, even though ECOBPG does not understand the composite type itself.

Finally, it is also possible to store composite type values in their external string representation in host variables of type VARCHAR. But that way, it is not easily possible to access the fields of the value from the host program.

User-defined Base Types

New user-defined base types are not directly supported by ECOBPG. You can use the external string representation and host variables of type VARCHAR, and this solution is indeed appropriate and sufficient for many types.

Here is an example using the data type complex.

See

Refer to "User-defined Types" in "Server Programming" in the PostgreSQL Documentation for information on the data type complex.

The external string representation of that type is (%lf,%lf), which is defined in the functions complex_in() and complex_out() functions. The following example inserts the complex type values (1,1) and (3,3) into the columns a and b, and select them from the table after that.

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
    01 A PIC X(64) VARYING.
    01 B PIC X(64) VARYING.
EXEC SQL END DECLARE SECTION END-EXEC.

EXEC SQL INSERT INTO test_complex VALUES ('(1,1)', '(3,3)') END-EXEC.

EXEC SQL DECLARE cur1 CURSOR FOR SELECT a, b FROM test_complex END-EXEC.
EXEC SQL OPEN cur1 END-EXEC.

EXEC SQL WHENEVER NOT FOUND GOTO END-FETCH END-EXEC.

PERFORM NO LIMIT
    EXEC SQL FETCH FROM cur1 INTO :A, :B END-EXEC
    DISPLAY "a=" ARR OF A ", b=" ARR OF B
END-PERFORM.

END-FETCH.
EXEC SQL CLOSE cur1 END-EXEC.
```

Another workaround is avoiding the direct use of the user-defined types in ECOBPG and instead create a function or cast that converts between the user-defined type and a primitive type that ECOBPG can handle. Note, however, that type casts, especially implicit ones, should be introduced into the type system very carefully.

For example:

```
CREATE FUNCTION create_complex(r double precision, i double precision) RETURNS complex
LANGUAGE SQL
IMMUTABLE
AS $$ SELECT $1 * complex '(1,0)' + $2 * complex '(0,1)' $$;
```

After this definition, the following:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 A COMP-2.
01 B COMP-2.
01 C COMP-2.
01 D COMP-2.
EXEC SQL END DECLARE SECTION END-EXEC.

MOVE 1 TO A.
MOVE 2 TO B.
MOVE 3 TO C.
MOVE 4 TO D.

EXEC SQL INSERT INTO test_complex VALUES (create_complex(:A, :B), create_complex(:C, :D))
END-EXEC.
```

has the same effect as

```
EXEC SQL INSERT INTO test_complex VALUES ('(1,2)', '(3,4)') END-EXEC.
```

D.4.6 Indicators

The examples above do not handle null values. In fact, the retrieval examples will raise an error if they fetch a null value from the database. To be able to pass null values to the database or retrieve null values from the database, you need to append a second host variable specification to each host variable that contains data. This second host variable is called the *indicator* and contains a flag that tells whether the datum is null, in which case the value of the real host variable is ignored. Here is an example that handles the retrieval of null values correctly:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 VAL PIC X(50) VARYING.
01 VAL_IND PIC S9(9) COMP-5.
```

```
EXEC SQL END DECLARE SECTION END-EXEC.

 ...

EXEC SQL SELECT b INTO :VAL :VAL_IND FROM test1 END-EXEC.
```

The indicator variable val_ind will be zero if the value was not null, and it will be negative if the value was null.

The indicator has another function: if the indicator value is positive, it means that the value is not null, but it was truncated when it was stored in the host variable.

D.5 Dynamic SQL

In many cases, the particular SQL statements that an application has to execute are known at the time the application is written. In some cases, however, the SQL statements are composed at run time or provided by an external source. In these cases you cannot embed the SQL statements directly into the COBOL source code, but there is a facility that allows you to call arbitrary SQL statements that you provide in a string variable.

D.5.1 Executing Statements without a Result Set

The simplest way to execute an arbitrary SQL statement is to use the command EXECUTE IMMEDIATE. For example:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 STMT PIC X(30) VARYING.
EXEC SQL END DECLARE SECTION END-EXEC.

MOVE "CREATE TABLE test1 (...);" TO ARR OF STMT.
COMPUTE LEN OF STMT = FUNCTION STORED-CHAR-LENGTH (ARR OF STMT).
EXEC SQL EXECUTE IMMEDIATE :STMT END-EXEC.
```

EXECUTE IMMEDIATE can be used for SQL statements that do not return a result set (e.g., DDL, INSERT, UPDATE, DELETE). You cannot execute statements that retrieve data (e.g., SELECT) this way. The next section describes how to do that.

D.5.2 Executing a Statement with Input Parameters

A more powerful way to execute arbitrary SQL statements is to prepare them once and execute the prepared statement as often as you like. It is also possible to prepare a generalized version of a statement and then execute specific versions of it by substituting parameters. When preparing the statement, write question marks where you want to substitute parameters later. For example:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 STMT PIC X(40) VARYING.
EXEC SQL END DECLARE SECTION END-EXEC.

MOVE "INSERT INTO test1 VALUES(?, ?);" TO ARR OF STMT.
COMPUTE LEN OF STMT = FUNCTION STORED-CHAR-LENGTH (ARR OF STMT).
EXEC SQL PREPARE MYSTMT FROM :STMT END-EXEC.
 ...
EXEC SQL EXECUTE MYSTMT USING 42, 'foobar' END-EXEC.
```

When you don't need the prepared statement anymore, you should deallocate it:

```
EXEC SQL DEALLOCATE PREPARE name END-EXEC.
```

D.5.3 Executing a Statement with a Result Set

To execute an SQL statement with a single result row, EXECUTE can be used. To save the result, add an INTO clause.

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 STMT PIC X(50) VARYING.
```

```
01 V1 PIC S9(9) COMP.
01 V2 PIC S9(9) COMP.
01 V3 PIC X(50) VARYING.
EXEC SQL END DECLARE SECTION END-EXEC.

MOVE "SELECT a, b, c FROM test1 WHERE a > ?" TO ARR OF STMT.
COMPUTE LEN OF STMT = FUNCTION STORED-CHAR-LENGTH (ARR OF STMT).
EXEC SQL PREPARE MYSTMT FROM :STMT END-EXEC.
 ...
EXEC SQL EXECUTE MYSTMT INTO :V1, :V2, :V3 USING 37 END-EXEC.
```

An EXECUTE command can have an INTO clause, a USING clause, both, or neither.

If a query is expected to return more than one result row, a cursor should be used, as in the following example. (See "D.3.2 Using Cursors" for more details about the cursor.)

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 DBANAME PIC X(128) VARYING.
01 DATNAME PIC X(128) VARYING.
01 STMT PIC X(200) VARYING.
EXEC SQL END DECLARE SECTION END-EXEC.

MOVE "SELECT u.usename as dbaname, d.datname
-             "  FROM pg_database d, pg_user u
-             "  WHERE d.datdba = u.usesysid"
TO ARR OF STMT.
COMPUTE LEN OF STMT = FUNCTION STORED-CHAR-LENGTH (ARR OF STMT).

EXEC SQL CONNECT TO testdb AS con1 USER testuser END-EXEC.

EXEC SQL PREPARE STMT1 FROM :STMT END-EXEC.

EXEC SQL DECLARE cursor1 CURSOR FOR STMT1 END-EXEC.
EXEC SQL OPEN cursor1 END-EXEC.

EXEC SQL WHENEVER NOT FOUND GOTO FETCH-END END-EXEC.

PERFORM NO LIMIT
    EXEC SQL FETCH cursor1 INTO :DBANAME,:DATNAME END-EXEC
    DISPLAY "dbaname=" ARR OF DBANAME ", datname=" ARR OF DATNAME
END-PERFORM.

FETCH-END.

EXEC SQL CLOSE cursor1 END-EXEC.

EXEC SQL COMMIT END-EXEC.
EXEC SQL DISCONNECT ALL END-EXEC.
```

D.6 Using Descriptor Areas

An SQL descriptor area is a more sophisticated method for processing the result of a SELECT, FETCH or a DESCRIBE statement. An SQL descriptor area groups the data of one row of data together with metadata items into one data group item. The metadata is particularly useful when executing dynamic SQL statements, where the nature of the result columns might not be known ahead of time. PostgreSQL provides a way to use Descriptor Areas: the named SQL Descriptor Areas.

D.6.1 Named SQL Descriptor Areas

A named SQL descriptor area consists of a header, which contains information concerning the entire descriptor, and one or more item descriptor areas, which basically each describe one column in the result row.

Before you can use an SQL descriptor area, you need to allocate one:

```
EXEC SQL ALLOCATE DESCRIPTOR identifier END-EXEC.
```

The identifier serves as the "variable name" of the descriptor area. When you don't need the descriptor anymore, you should deallocate it:

```
EXEC SQL DEALLOCATE DESCRIPTOR identifier END-EXEC.
```

To use a descriptor area, specify it as the storage target in an INTO clause, instead of listing host variables:

```
EXEC SQL FETCH NEXT FROM mycursor INTO SQL DESCRIPTOR mydesc END-EXEC.
```

If the result set is empty, the Descriptor Area will still contain the metadata from the query, i.e. the field names.

For not yet executed prepared queries, the DESCRIBE statement can be used to get the metadata of the result set:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 SQL-STMT PIC X(30) VARYING.
EXEC SQL END DECLARE SECTION END-EXEC.

MOVE "SELECT * FROM table1" TO ARR OF SQL-STMT.
COMPUTE LEN OF SQL-STMT = FUNCTION STORED-CHAR-LENGTH ( ARR OF SQL-STMT ) .
EXEC SQL PREPARE STMT1 FROM :SQL-STMT END-EXEC.
EXEC SQL DESCRIBE STMT1 INTO SQL DESCRIPTOR MYDESC END-EXEC.
```

Before PostgreSQL 9.0, the SQL keyword was optional, so using DESCRIPTOR and SQL DESCRIPTOR produced named SQL Descriptor Areas. Now it is mandatory, omitting the SQL keyword is regarded as the syntax that produces SQLDA Descriptor Areas. However, ecobpg does not support SQLDA and it causes an error.

In DESCRIBE and FETCH statements, the INTO and USING keywords can be used to similarly: they produce the result set and the metadata in a Descriptor Area.

Now how do you get the data out of the descriptor area? You can think of the descriptor area as a group item with named fields. To retrieve the value of a field from the header and store it into a host variable, use the following command:

```
EXEC SQL GET DESCRIPTOR name :hostvar = field END-EXEC.
```

Currently, there is only one header field defined: COUNT, which tells how many item descriptor areas exist (that is, how many columns are contained in the result). The host variable needs to be of an integer type as PIC S9(9) COMP-5. To get a field from the item descriptor area, use the following command:

```
EXEC SQL GET DESCRIPTOR name VALUE num :hostvar = field END-EXEC.
```

num can be a host variable containing an integer as PIC S9(9) COMP-5.
hostvar must be PIC S9(9) COMP-5 if type of the field is integer. Possible fields are:

CARDINALITY (integer)

number of rows in the result set

DATA

actual data item (therefore, the data type of this field depends on the query)

DATETIME_INTERVAL_CODE (integer)

When TYPE is 9, DATETIME_INTERVAL_CODE will have a value of 1 for DATE, 2 for TIME, 3 for TIMESTAMP, 4 for TIME WITH TIME ZONE, or 5 for TIMESTAMP WITH TIME ZONE.

DATETIME_INTERVAL_PRECISION (integer)

not implemented

INDICATOR (integer)

the indicator (indicating a null value or a value truncation)

KEY_MEMBER (integer)

not implemented

LENGTH (integer)

length of the datum in characters

NAME (string)

name of the column

NULLABLE (integer)

not implemented

OCTET_LENGTH (integer)

length of the character representation of the datum in bytes

PRECISION (integer)

precision (for type numeric)

RETURNED_LENGTH (integer)

length of the datum in characters

RETURNED_OCTET_LENGTH (integer)

length of the character representation of the datum in bytes

SCALE (integer)

scale (for type numeric)

TYPE (integer)

numeric code of the data type of the column

In EXECUTE, DECLARE and OPEN statements, the effect of the INTO and USING keywords are different. A Descriptor Area can also be manually built to provide the input parameters for a query or a cursor and USING SQL DESCRIPTOR name is the way to pass the input parameters into a parametrized query. The statement to build a named SQL Descriptor Area is below:

```
EXEC SQL SET DESCRIPTOR name VALUE num field = :hostvar END-EXEC.
```

PostgreSQL supports retrieving more than one record in one FETCH statement and storing the data in host variables in this case assumes that the variable is an array. E.g.:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 GROUP-ITEM.
    05 IDNUM PIC S9(9) COMP OCCURS 5.
EXEC SQL END DECLARE SECTION END-EXEC.

EXEC SQL FETCH 5 FROM mycursor INTO SQL DESCRIPTOR mydesc END-EXEC.

EXEC SQL GET DESCRIPTOR mydesc VALUE 1 :IDNUM = DATA END-EXEC.
```

D.7 Error Handling

This section describes how you can handle exceptional conditions and warnings in an embedded SQL program. There are two nonexclusive facilities for this.

- Callbacks can be configured to handle warning and error conditions using the WHENEVER command.
- Detailed information about the error or warning can be obtained from the sqlca variable.

D.7.1 Setting Callbacks

One simple method to catch errors and warnings is to set a specific action to be executed whenever a particular condition occurs. In general:

```
EXEC SQL WHENEVER condition action END-EXEC.
```

condition can be one of the following:

SQLERROR

The specified action is called whenever an error occurs during the execution of an SQL statement.

SQLWARNING

The specified action is called whenever a warning occurs during the execution of an SQL statement.

NOT FOUND

The specified action is called whenever an SQL statement retrieves or affects zero rows. (This condition is not an error, but you might be interested in handling it specially.)

action can be one of the following:

CONTINUE

This effectively means that the condition is ignored. This is the default.

GOTO label
GO TO label

Jump to the specified label (using a COBOL goto statement).

SQLPRINT

Print a message to standard error. This is useful for simple programs or during prototyping. The details of the message cannot be configured.

STOP

Call STOP, which will terminate the program.

CALL name usingargs
DO name usingargs

Call the specified functions with the following characters including arguments. Thus, syntaxes (including compiler depending) are able to be placed as well as the arguments. Though, there are some limitation as following:

- You can't use RETURNING, ON EXCEPTION or OVER FLOW clauses.
- In the called subprogram, you must specify CONTINUE for every action with WHENEVER statement.

The SQL standard only provides for the actions CONTINUE and GOTO (and GO TO).

Here is an example that you might want to use in a simple program. It prints a simple message when a warning occurs and aborts the program when an error happens:

```
EXEC SQL WHENEVER SQLWARNING SQLPRINT END-EXEC.
EXEC SQL WHENEVER SQLERROR STOP END-EXEC.
```

The statement EXEC SQL WHENEVER is a directive of the SQL preprocessor, not a COBOL statement. The error or warning actions that it sets apply to all embedded SQL statements that appear below the point where the handler is set, unless a different action was set for the same condition between the first EXEC SQL WHENEVER and the SQL statement causing the condition, regardless of the flow of control in the COBOL program. So neither of the two following COBOL program excerpts will have the desired effect:

```
*
*   WRONG
*
    ...
    IF VERBOSE = 1 THEN
        EXEC SQL WHENEVER SQLWARNING SQLPRINT END-EXEC
    END-IF.
```

```
    ...
    EXEC SQL SELECT ... END-EXEC.
    ...
*
*   WRONG
*
    ...
    CALL SET-ERROR-HANDLER.
*        (and execute "EXEC SQL WHENEVER SQLERROR STOP" in SET-ERROR-HANDLER)
    ...
    EXEC SQL SELECT ... END-EXEC.
    ...
```

D.7.2 sqlca

For more powerful error handling, the embedded SQL interface provides a global variable with the name sqlca (SQL communication area) that has the following group item:

```
01  sqlca_t.
    10 sqlcaid PIC X(8).
    10 sqlabc PIC S9(9) COMP-5.
    10 sqlcode PIC S9(9) COMP-5.
    10 sqlerrm.
        20 sqlerrml PIC S9(9) COMP-5.
        20 sqlerrmc PIC X(150).
    10 sqlerrp PIC X(8).
    10 sqlerrd PIC S9(9) COMP-5 OCCURS 6.
    10 sqlwarn PIC X(8).
    10 sqlstate PIC X(5).
```

sqlca covers both warnings and errors. If multiple warnings or errors occur during the execution of a statement, then sqlca will only contain information about the last one.

If no error occurred in the last SQL statement, SQLCODE will be 0 and SQLSTATE will be "00000". If a warning or error occurred, then SQLCODE will be negative and SQLSTATE will be different from "00000". A positive SQLCODE indicates a harmless condition, such as that the last query returned zero rows. SQLCODE and SQLSTATE are two different error code schemes; details appear below.

If the last SQL statement was successful, then SQLERRD(2) contains the OID of the processed row, if applicable, and SQLERRD(3) contains the number of processed or returned rows, if applicable to the command.

In case of an error or warning, SQLERRMC will contain a string that describes the error. The field SQLERRML contains the length of the error message that is stored in SQLERRMC (the result of FUNCTION STORED-CHAR-LENGTH. Note that some messages are too long to fit in the fixed-size sqlerrmc array; they will be truncated.

In case of a warning, the 3rd character of SQLWARN is set to W. (In all other cases, it is set to something different from W.) If the 2nd character of SQLWARN is set to W, then a value was truncated when it was stored in a host variable. The 1st character of SQLWARN is set to W if any of the other elements are set to indicate a warning.

The fields sqlcaid, sqlcabc, sqlerrp, and the remaining elements of sqlerrd and sqlwarn currently contain no useful information.

The structure sqlca is not defined in the SQL standard, but is implemented in several other SQL database systems. The definitions are similar at the core, but if you want to write portable applications, then you should investigate the different implementations carefully.

Here is one example that combines the use of WHENEVER and sqlca, printing out the contents of sqlca when an error occurs. This is perhaps useful for debugging or prototyping applications, before installing a more "user-friendly" error handler.

```
EXEC SQL WHENEVER SQLERROR GOTO PRINT_SQLCA END-EXEC.

PRINT_SQLCA.
    DISPLAY "==== sqlca ====".
    DISPLAY "SQLCODE: " SQLCODE.
```

```
    DISPLAY "SQLERRML: " SQLERRML.
    DISPLAY "SQLERRMC: " SQLERRMC.
    DISPLAY "SQLERRD: " SQLERRD(1) " " SQLERRD(2) " " SQLERRD(3)" " SQLERRD(4) " "
SQLERRD(5) " " SQLERRD(6).
    DISPLAY "SQLSTATE: " SQLSTATE.
    DISPLAY "===============".
```

The result could look as follows (here an error due to a misspelled table name):

```
==== sqlca ====
sqlcode: -000000400
SQLERRML: +000000064
SQLERRMC: relation "pg_databasep" does not exist (10292) on line 93
sqlerrd: +000000000 +000000000 +000000000 +000000000 +000000000 +000000000
sqlstate: 42P01
===============
```

D.7.3 SQLSTATE vs. SQLCODE

The fields SQLSTATE and SQLCODE are two different schemes that provide error codes. Both are derived from the SQL standard, but SQLCODE has been marked deprecated in the SQL-92 edition of the standard and has been dropped in later editions. Therefore, new applications are strongly encouraged to use SQLSTATE.

SQLSTATE is a five-character array. The five characters contain digits or upper-case letters that represent codes of various error and warning conditions. SQLSTATE has a hierarchical scheme: the first two characters indicate the general class of the condition, the last three characters indicate a subclass of the general condition. A successful state is indicated by the code 00000. The SQLSTATE codes are for the most part defined in the SQL standard. The PostgreSQL server natively supports SQLSTATE error codes; therefore a high degree of consistency can be achieved by using this error code scheme throughout all applications.

See

Refer to "PostgreSQL Error Codes" in "Appendixes" in the PostgreSQL Documentation for further information.

SQLCODE, the deprecated error code scheme, is a simple integer. A value of 0 indicates success, a positive value indicates success with additional information, and a negative value indicates an error. The SQL standard only defines the positive value +100, which indicates that the last command returned or affected zero rows, and no specific negative values. Therefore, this scheme can only achieve poor portability and does not have a hierarchical code assignment. Historically, the embedded SQL processor for PostgreSQL has assigned some specific SQLCODE values for its use, which are listed below with their numeric value and their symbolic name. Remember that these are not portable to other SQL implementations. To simplify the porting of applications to the SQLSTATE scheme, the corresponding SQLSTATE is also listed. There is, however, no one-to-one or one-to-many mapping between the two schemes (indeed it is many-to-many), so you should consult the global SQLSTATE in each case.

See

Refer to "PostgreSQL Error Codes" in "Appendixes" in the PostgreSQL Documentation.

These are the assigned SQLCODE values:

0

Indicates no error. (SQLSTATE 00000)

100

This is a harmless condition indicating that the last command retrieved or processed zero rows, or that you are at the end of the cursor. (SQLSTATE 02000)

When processing a cursor in a loop, you could use this code as a way to detect when to abort the loop, like this:

```
PERFORM NO LIMIT
    EXEC SQL FETCH ... END-EXEC
    IF SQLCODE = 100 THEN
        GO TO FETCH-END
    END-IF
END-PERFORM.
```

But WHENEVER NOT FOUND GOTO ... effectively does this internally, so there is usually no advantage in writing this out explicitly.

-12

Indicates that your virtual memory is exhausted. The numeric value is defined as -ENOMEM. (SQLSTATE YE001)

-200

Indicates the preprocessor has generated something that the library does not know about. Perhaps you are running incompatible versions of the preprocessor and the library. (SQLSTATE YE002)

-201

This means that the command specified more host variables than the command expected. (SQLSTATE 07001 or 07002)

-202

This means that the command specified fewer host variables than the command expected. (SQLSTATE 07001 or 07002)

-203

This means a query has returned multiple rows but the statement was only prepared to store one result row (for example, because the specified variables are not arrays). (SQLSTATE 21000)

-204

The host variable is of type signed int and the datum in the database is of a different type and contains a value that cannot be interpreted as a signed int. The library uses strtol() for this conversion. (SQLSTATE 42804)

-205

The host variable is of type unsigned int and the datum in the database is of a different type and contains a value that cannot be interpreted as an unsigned int. The library uses strtoul() for this conversion. (SQLSTATE 42804)

-206

The host variable is of type float and the datum in the database is of another type and contains a value that cannot be interpreted as a float. The library uses strtod() for this conversion. (SQLSTATE 42804)

-207

The host variable is of type DECIMAL and the datum in the database is of another type and contains a value that cannot be interpreted as a DECIMAL or DISPLAY value. For the case of DISPLAY, this error happens if values in the database is too large for converting to DISPLAY value. (SQLSTATE 42804)

-208

The host variable is of type interval and the datum in the database is of another type and contains a value that cannot be interpreted as an interval value. (SQLSTATE 42804)

-209

The host variable is of type date and the datum in the database is of another type and contains a value that cannot be interpreted as a date value. (SQLSTATE 42804)

-210

The host variable is of type timestamp and the datum in the database is of another type and contains a value that cannot be interpreted as a timestamp value. (SQLSTATE 42804)

-211

This means the host variable is of type bool and the datum in the database is neither 't' nor 'f'. (SQLSTATE 42804)

-212

The statement sent to the PostgreSQL server was empty. (This cannot normally happen in an embedded SQL program, so it might point to an internal error.) (SQLSTATE YE002)

-213

A null value was returned and no null indicator variable was supplied. (SQLSTATE 22002)

-214

An ordinary variable was used in a place that requires an array. (SQLSTATE 42804)

-215

The database returned an ordinary variable in a place that requires array value. (SQLSTATE 42804)

-220

The program tried to access a connection that does not exist. (SQLSTATE 08003)

-221

The program tried to access a connection that does exist but is not open. (This is an internal error.) (SQLSTATE YE002)

-230

The statement you are trying to use has not been prepared. (SQLSTATE 26000)

-240

The descriptor specified was not found. The statement you are trying to use has not been prepared. (SQLSTATE 33000)

-241

The descriptor index specified was out of range. (SQLSTATE 07009)

-242

An invalid descriptor item was requested. (This is an internal error.) (SQLSTATE YE002)

-243

During the execution of a dynamic statement, the database returned a numeric value and the host variable was not numeric. (SQLSTATE 07006)

-244

During the execution of a dynamic statement, the database returned a non-numeric value and the host variable was numeric. (SQLSTATE 07006)

-400

Some error caused by the PostgreSQL server. The message contains the error message from the PostgreSQL server.

-401

The PostgreSQL server signaled that we cannot start, commit, or rollback the transaction. (SQLSTATE 08007)

-402

The connection attempt to the database did not succeed. (SQLSTATE 08001)

-403

Duplicate key error, violation of unique constraint. (SQLSTATE 23505)

-404

A result for the subquery is not single row. (SQLSTATE 21000)

-602

An invalid cursor name was specified. (SQLSTATE 34000)

-603

Transaction is in progress. (SQLSTATE 25001)

-604

There is no active (in-progress) transaction. (SQLSTATE 25P01)

-605

An existing cursor name was specified. (SQLSTATE 42P03)

D.8 Preprocessor Directives

Several preprocessor directives are available that modify how the ecobpg preprocessor parses and processes a file.

D.8.1 Including Files

To include an external file into your embedded SQL program, use:

```
EXEC SQL INCLUDE filename END-EXEC.
EXEC SQL INCLUDE <filename> END-EXEC.
EXEC SQL INCLUDE "filename" END-EXEC.
```

The embedded SQL preprocessor will look for a file named filename.pco, preprocess it, and include it in the resulting COBOL output. Thus, embedded SQL statements in the included file are handled correctly.

By default, the ecobpg preprocessor will search a file at the current directory. This behavior can be changed by the ecobpg commandline option.

First, the preprocessor tries to locate a file by specified file name at the current directory. If it fails and the file name does not end with .pco, the preprocessor also tries to locate a file with the suffix at the same directory.

The difference between EXEC SQL INCLUDE and COPY statement is whether precompiler processes embedded SQLs in the file, or not. If the file contains embedded SQLs, use EXEC SQL INCLUDE.

Note

The include file name is case-sensitive, even though the rest of the EXEC SQL INCLUDE command follows the normal SQL case-sensitivity rules.

D.8.2 The define and undef Directives

Similar to the directive #define that is known from C, embedded SQL has a similar concept:

```
EXEC SQL DEFINE name END-EXEC.
EXEC SQL DEFINE name value END-EXEC.
```

So you can define a name:

```
EXEC SQL DEFINE HAVE_FEATURE END-EXEC.
```

And you can also define constants:

```
EXEC SQL DEFINE MYNUMBER 12 END-EXEC.
EXEC SQL DEFINE MYSTRING 'abc' END-EXEC.
```

Use undef to remove a previous definition:

```
EXEC SQL UNDEF MYNUMBER END-EXEC.
```

Note that a constant in the SQL statement is only replaced by EXEC SQL DEFINE. The replacement may change the number of characters in a line, but ecobpg does not validate it after the replacement. Pay attention to the limitation of the number of characters in a line.

D.8.3 ifdef, ifndef, else, elif, and endif Directives

You can use the following directives to compile code sections conditionally:

```
EXEC SQL ifdef name END-EXEC.
```

Checks a name and processes subsequent lines if name has been created with EXEC SQL define name.

```
EXEC SQL ifndef name END-EXEC.
```

Checks a name and processes subsequent lines if name has not been created with EXEC SQL define name.

```
EXEC SQL else END-EXEC.
```

Starts processing an alternative section to a section introduced by either EXEC SQL ifdef name or EXEC SQL ifndef name.

```
EXEC SQL elif name END-EXEC.
```

Checks name and starts an alternative section if name has been created with EXEC SQL define name.

```
EXEC SQL endif END-EXEC.
```

Ends an alternative section.

Example:

```
EXEC SQL ifndef TZVAR END-EXEC.
EXEC SQL SET TIMEZONE TO 'GMT' END-EXEC.
EXEC SQL elif TZNAME END-EXEC.
EXEC SQL SET TIMEZONE TO TZNAME END-EXEC.
EXEC SQL else END-EXEC.
EXEC SQL SET TIMEZONE TO TZVAR END-EXEC.
EXEC SQL endif END-EXEC.
```

D.9 Processing Embedded SQL Programs

Now that you have an idea how to form embedded SQL COBOL programs, you probably want to know how to compile them. Before compiling you run the file through the embedded SQL COBOL preprocessor, which converts the SQL statements you used to special function calls. After compiling, you must link with a special library that contains the needed functions. These functions fetch information from the arguments, perform the SQL command using the libpq interface, and put the result in the arguments specified for output.

The preprocessor program is called ecobpg and is included in a normal PostgreSQL installation. Embedded SQL programs are typically named with an extension .pco. If you have a program file called prog1.pco, you can preprocess it by simply calling:

```
ecobpg prog1.pco
```

This will create a file called prog1.cob. If your input files do not follow the suggested naming pattern, you can specify the output file explicitly using the -o option.

The preprocessed file can be compiled normally, following the usage of the compiler.

The generated COBOL source files include library files from the PostgreSQL installation, so if you installed PostgreSQL in a location that is not searched by default, you have to add an option such as -I/usr/local/pgsql/include to the compilation command line.

To link an embedded SQL program, you need to include the libecpg library.

Again, you might have to add an option for library search like -L/usr/local/pgsql/lib to that command line.

If you manage the build process of a larger project using make, it might be convenient to include the following implicit rule to your makefiles:

```
ECOBPG = ecobpg

%.cob: %.pco
        $(ECOBPG) $<
```

The complete syntax of the ecobpg command is detailed in "D.12.1 ecobpg".

Currently, ecobpg does not support multi threading.

D.10 Large Objects

Large objects are not supported by ECOBPG.

If you need to access large objects, use large objects interfaces of libpq instead.

D.11 Embedded SQL Commands

This section describes all SQL commands that are specific to embedded SQL.

Command	Description
ALLOCATE DESCRIPTOR	Allocate an SQL descriptor area
CONNECT	Establish a database connection
DEALLOCATE DESCRIPTOR	Deallocate an SQL descriptor area
DECLARE	Define a cursor
DESCRIBE	Obtain information about a prepared statement or result set
DISCONNECT	Terminate a database connection
EXECUTE IMMEDIATE	Dynamically prepare and execute a statement
GET DESCRIPTOR	Get information from an SQL descriptor area
OPEN	Open a dynamic cursor
PREPARE	Prepare a statement for execution
SET AUTOCOMMIT	Set the autocommit behavior of the current session
SET CONNECTION	Select a database connection
SET DESCRIPTOR	Set information in an SQL descriptor area
TYPE	Define a new data type
VAR	Define a variable
WHENEVER	Specify the action to be taken when an SQL statement causes a specific class condition to be raised

Refer to the SQL commands listed in "SQL Commands" under "Reference" in the PostgreSQL Documentation, which can also be used in embedded SQL, unless stated otherwise.

D.11.1 ALLOCATE DESCRIPTOR

Name

ALLOCATE DESCRIPTOR -- allocate an SQL descriptor area

Synopsis

```
ALLOCATE DESCRIPTOR name
```

Description

ALLOCATE DESCRIPTOR allocates a new named SQL descriptor area, which can be used to exchange data between the PostgreSQL server and the host program.

Descriptor areas should be freed after use using the DEALLOCATE DESCRIPTOR command.

Parameters

name

A name of SQL descriptor. This can be an SQL identifier or a host variable.

Examples

```
EXEC SQL ALLOCATE DESCRIPTOR mydesc END-EXEC.
```

Compatibility

ALLOCATE DESCRIPTOR is specified in the SQL standard.

See Also

DEALLOCATE DESCRIPTOR, GET DESCRIPTOR, SET DESCRIPTOR

D.11.2 CONNECT

Name

CONNECT -- establish a database connection

Synopsis

```
CONNECT TO connection_target [ AS connection_name ] [ USER connection_user_name ]

CONNECT TO DEFAULT

CONNECT connection_user_name

DATABASE connection_target
```

Description

The CONNECT command establishes a connection between the client and the PostgreSQL server.

Parameters

connection_target

connection_target specifies the target server of the connection on one of several forms.

[database_name] [@host] [:port]

Connect over TCP/IP

unix:postgresql://host [:port] / [database_name] [?connection_option]

Connect over Unix-domain sockets

tcp:postgresql://host [:port] / [database_name] [?connection_option]

Connect over TCP/IP

SQL string constant

containing a value in one of the above forms

host variable

host variable of fixed-length string (trailing spaces are ignored) containing a value in one of the above forms

connection_name

An optional identifier for the connection, so that it can be referred to in other commands. This can be an SQL identifier or a host variable.

connection_user_name

The user name for the database connection.

This parameter can also specify user name and password, using one the forms user_name/password, user_name IDENTIFIED BY password, or user_name USING password.

User name and password can be SQL identifiers, string constants, or host variables (fixed-length string, trailing spaces are ignored).

DEFAULT

Use all default connection parameters, as defined by libpq.

Examples

Here a several variants for specifying connection parameters:

```
EXEC SQL CONNECT TO "connectdb" AS main END-EXEC.
EXEC SQL CONNECT TO "connectdb" AS second END-EXEC.
EXEC SQL CONNECT TO "unix:postgresql://localhost/connectdb" AS main USER connectuser END-
EXEC.
EXEC SQL CONNECT TO 'connectdb' AS main END-EXEC.
EXEC SQL CONNECT TO 'unix:postgresql://localhost/connectdb' AS main USER :user END-EXEC.
EXEC SQL CONNECT TO :dbn AS :idt END-EXEC.
EXEC SQL CONNECT TO :dbn USER connectuser USING :pw END-EXEC.
EXEC SQL CONNECT TO @localhost AS main USER connectdb END-EXEC.
EXEC SQL CONNECT TO REGRESSDB1 as main END-EXEC.
EXEC SQL CONNECT TO connectdb AS :idt END-EXEC.
EXEC SQL CONNECT TO connectdb AS main USER connectuser/connectdb END-EXEC.
EXEC SQL CONNECT TO connectdb AS main END-EXEC.
EXEC SQL CONNECT TO connectdb@localhost AS main END-EXEC.
EXEC SQL CONNECT TO tcp:postgresql://localhost/ USER connectdb END-EXEC.
EXEC SQL CONNECT TO tcp:postgresql://localhost/connectdb USER connectuser IDENTIFIED BY
connectpw END-EXEC.
EXEC SQL CONNECT TO tcp:postgresql://localhost:20/connectdb USER connectuser IDENTIFIED BY
connectpw END-EXEC.
EXEC SQL CONNECT TO unix:postgresql://localhost/ AS main USER connectdb END-EXEC.
EXEC SQL CONNECT TO unix:postgresql://localhost/connectdb AS main USER connectuser END-
EXEC.
EXEC SQL CONNECT TO unix:postgresql://localhost/connectdb USER connectuser IDENTIFIED BY
"connectpw" END-EXEC.
EXEC SQL CONNECT TO unix:postgresql://localhost/connectdb USER connectuser USING
"connectpw" END-EXEC.
EXEC SQL CONNECT TO unix:postgresql://localhost/connectdb?connect_timeout=14 USER
connectuser END-EXEC.
```

Here is an example program that illustrates the use of host variables to specify connection parameters:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
*   database name
    01 DBNAME PIC X(6).
*   connection user name
```

```
    01 USER PIC X(8).
*   connection string
    01 CONNECTION PIC X(38).

    01 VER PIC X(256).
EXEC SQL END DECLARE SECTION END-EXEC.

    MOVE "testdb" TO DBNAME.
    MOVE "testuser" TO USER.
    MOVE "tcp:postgresql://localhost:5432/testdb" TO CONNECTION.

    EXEC SQL CONNECT TO :DBNAME USER :USER END-EXEC.
    EXEC SQL SELECT version() INTO :VER END-EXEC.
    EXEC SQL DISCONNECT END-EXEC.


    DISPLAY "version: " VER.

    EXEC SQL CONNECT TO :CONNECTION USER :USER END-EXEC.
    EXEC SQL SELECT version() INTO :VER END-EXEC.
    EXEC SQL DISCONNECT END-EXEC.

    DISPLAY "version: " VER.
```

Compatibility

CONNECT is specified in the SQL standard, but the format of the connection parameters is implementation-specific.

See Also

DISCONNECT, SET CONNECTION

D.11.3 DEALLOCATE DESCRIPTOR

Name

DEALLOCATE DESCRIPTOR -- deallocate an SQL descriptor area

Synopsis

```
DEALLOCATE DESCRIPTOR name
```

Description

DEALLOCATE DESCRIPTOR deallocates a named SQL descriptor area.

Parameters

name

The name of the descriptor which is going to be deallocated. This can be an SQL identifier or a host variable.

Examples

EXEC SQL DEALLOCATE DESCRIPTOR mydesc END-EXEC.

Compatibility

DEALLOCATE DESCRIPTOR is specified in the SQL standard.

See Also

ALLOCATE DESCRIPTOR, GET DESCRIPTOR, SET DESCRIPTOR

D.11.4 DECLARE

Name

DECLARE -- define a cursor

Synopsis

```
DECLARE cursor_name [ BINARY ] [ INSENSITIVE ] [ [ NO ] SCROLL ] CURSOR [ { WITH | WITHOUT }
HOLD ] FOR prepared_name

DECLARE cursor_name [ BINARY ] [ INSENSITIVE ] [ [ NO ] SCROLL ] CURSOR [ { WITH | WITHOUT }
HOLD ] FOR query
```

Description

DECLARE declares a cursor for iterating over the result set of a prepared statement. This command has slightly different semantics from the direct SQL command DECLARE: Whereas the latter executes a query and prepares the result set for retrieval, this embedded SQL command merely declares a name as a "loop variable" for iterating over the result set of a query; the actual execution happens when the cursor is opened with the OPEN command.

Parameters

cursor_name

A cursor name. This can be an SQL identifier or a host variable.

prepared_name

The name of a prepared query, either as an SQL identifier or a host variable.

query

A SELECT or VALUES command which will provide the rows to be returned by the cursor.

For the meaning of the cursor options, see DECLARE.

Refer to "SQL Commands" in "Reference" in the PostgreSQL Documentation for information on the SELECT, VALUES and DECLARE command.

Examples

Examples declaring a cursor for a query:

```
EXEC SQL DECLARE C CURSOR FOR SELECT * FROM My_Table END-EXEC.
EXEC SQL DECLARE C CURSOR FOR SELECT Item1 FROM T END-EXEC.
EXEC SQL DECLARE cur1 CURSOR FOR SELECT version() END-EXEC.
```

An example declaring a cursor for a prepared statement:

```
EXEC SQL PREPARE stmt1 AS SELECT version() END-EXEC.
EXEC SQL DECLARE cur1 CURSOR FOR stmt1 END-EXEC.
```

Compatibility

DECLARE is specified in the SQL standard.

See Also

OPEN, CLOSE, DECLARE

See

Refer to "SQL Commands" in "Reference" in the PostgreSQL Documentation for information on the CLOSE and DECLARE command.

D.11.5 DESCRIBE

Name

DESCRIBE -- obtain information about a prepared statement or result set

Synopsis

```
DESCRIBE [ OUTPUT ] prepared_name USING SQL DESCRIPTOR descriptor_name

DESCRIBE [ OUTPUT ] prepared_name INTO SQL DESCRIPTOR descriptor_name
```

Description

DESCRIBE retrieves metadata information about the result columns contained in a prepared statement, without actually fetching a row.

Parameters

prepared_name

The name of a prepared statement. This can be an SQL identifier or a host variable.

descriptor_name

A descriptor name. It can be an SQL identifier or a host variable.

Examples

```
EXEC SQL ALLOCATE DESCRIPTOR mydesc END-EXEC.
EXEC SQL PREPARE stmt1 FROM :sql_stmt END-EXEC.
EXEC SQL DESCRIBE stmt1 INTO SQL DESCRIPTOR mydesc END-EXEC.
EXEC SQL GET DESCRIPTOR mydesc VALUE 1 :charvar = NAME END-EXEC.
EXEC SQL DEALLOCATE DESCRIPTOR mydesc END-EXEC.
```

Compatibility

DESCRIBE is specified in the SQL standard.

See Also

ALLOCATE DESCRIPTOR, GET DESCRIPTOR

D.11.6 DISCONNECT

Name

DISCONNECT -- terminate a database connection

Synopsis

```
DISCONNECT connection_name

DISCONNECT [ CURRENT ]

DISCONNECT DEFAULT
```

```
DISCONNECT ALL
```

Description

DISCONNECT closes a connection (or all connections) to the database.

Parameters

connection_name

A database connection name established by the CONNECT command.

CURRENT

Close the "current" connection, which is either the most recently opened connection, or the connection set by the SET CONNECTION command. This is also the default if no argument is given to the DISCONNECT command.

DEFAULT

Close the default connection.

ALL

Close all open connections.

Examples

```
    EXEC SQL CONNECT TO testdb AS DEFAULT USER testuser END-EXEC.
    EXEC SQL CONNECT TO testdb AS con1 USER testuser END-EXEC.
    EXEC SQL CONNECT TO testdb AS con2 USER testuser END-EXEC.
    EXEC SQL CONNECT TO testdb AS con3 USER testuser END-EXEC.

*   close con3
    EXEC SQL DISCONNECT CURRENT END-EXEC.
*   close DEFAULT
    EXEC SQL DISCONNECT DEFAULT END-EXEC.
*   close con2 and con1
    EXEC SQL DISCONNECT ALL END-EXEC.
```

Compatibility

DISCONNECT is specified in the SQL standard.

See Also

CONNECT, SET CONNECTION

D.11.7 EXECUTE IMMEDIATE

Name

EXECUTE IMMEDIATE -- dynamically prepare and execute a statement

Synopsis

```
EXECUTE IMMEDIATE string
```

Description

EXECUTE IMMEDIATE immediately prepares and executes a dynamically specified SQL statement, without retrieving result rows.

Parameters

string

A literal string or a host variable containing the SQL statement to be executed.

Examples

Here is an example that executes an INSERT statement using EXECUTE IMMEDIATE and a host variable named command:

```
MOVE "INSERT INTO test (name, amount, letter) VALUES ('db: ''r1''', 1, 'f')" TO ARR OF cmd.
COMPUTE LEN OF cmd = FUNCTION STORED-CHAR-LENGTH(ARR OF cmd).
EXEC SQL EXECUTE IMMEDIATE :cmd END-EXEC.
```

Compatibility

EXECUTE IMMEDIATE is specified in the SQL standard.

D.11.8 GET DESCRIPTOR

Name

GET DESCRIPTOR -- get information from an SQL descriptor area

Synopsis

```
GET DESCRIPTOR descriptor_name :hostvariable = descriptor_header_item [, ... ]

GET DESCRIPTOR descriptor_name VALUE column_number :hostvariable = descriptor_item [, ... ]
```

Description

GET DESCRIPTOR retrieves information about a query result set from an SQL descriptor area and stores it into host variables. A descriptor area is typically populated using FETCH or SELECT before using this command to transfer the information into host language variables.

This command has two forms: The first form retrieves descriptor "header" items, which apply to the result set in its entirety. One example is the row count. The second form, which requires the column number as additional parameter, retrieves information about a particular column. Examples are the column name and the actual column value.

Parameters

descriptor_name

A descriptor name.

descriptor_header_item

A token identifying which header information item to retrieve. Only COUNT, to get the number of columns in the result set, is currently supported.

column_number

The number of the column about which information is to be retrieved. The count starts at 1.

descriptor_item

A token identifying which item of information about a column to retrieve. See Section 33.7.1 for a list of supported items.

hostvariable

A host variable that will receive the data retrieved from the descriptor area.

Examples

An example to retrieve the number of columns in a result set:

```
EXEC SQL GET DESCRIPTOR d :d_count = COUNT END-EXEC.
```

An example to retrieve a data length in the first column:

```
EXEC SQL GET DESCRIPTOR d VALUE 1 :d_returned_octet_length = RETURNED_OCTET_LENGTH END-
EXEC.
```

An example to retrieve the data body of the second column as a string:

```
EXEC SQL GET DESCRIPTOR d VALUE 2 :d_data = DATA END-EXEC.
```

Here is an example for a whole procedure of executing SELECT current_database(); and showing the number of columns, the column data length, and the column data:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
    01 D-COUNT PIC S9(9) COMP-5.
    01 D-DATA PIC X(1024).
    01 D-RETURNED-OCTET-LENGTH PIC S9(9) COMP.
EXEC SQL END DECLARE SECTION END-EXEC.

    EXEC SQL CONNECT TO testdb AS con1 USER testuser END-EXEC.
    EXEC SQL ALLOCATE DESCRIPTOR d END-EXEC.

*   Declare, open a cursor, and assign a descriptor to the cursor
    EXEC SQL DECLARE cur CURSOR FOR SELECT current_database() END-EXEC.
    EXEC SQL OPEN cur END-EXEC.
    EXEC SQL FETCH NEXT FROM cur INTO SQL DESCRIPTOR d END-EXEC.

*   Get a number of total columns
    EXEC SQL GET DESCRIPTOR d :D-COUNT = COUNT END-EXEC.
    DISPLAY "d_count                = " D-COUNT.

*   Get length of a returned column
    EXEC SQL GET DESCRIPTOR d VALUE 1 :D-RETURNED-OCTET-LENGTH = RETURNED_OCTET_LENGTH END-
EXEC.
    DISPLAY "d_returned_octet_length = " D-RETURNED-OCTET-LENGTH.

*   Fetch the returned column as a string
    EXEC SQL GET DESCRIPTOR d VALUE 1 :D-DATA = DATA END-EXEC.
    DISPLAY "d_data                 = " D-DATA.

*   Closing
    EXEC SQL CLOSE cur END-EXEC.
    EXEC SQL COMMIT END-EXEC.

    EXEC SQL DEALLOCATE DESCRIPTOR d END-EXEC.
    EXEC SQL DISCONNECT ALL END-EXEC.
```

When the example is executed, the result will look like this:

```
d_count                = +000000001
d_returned_octet_length = +000000006
d_data                 = testdb
```

Compatibility

GET DESCRIPTOR is specified in the SQL standard.

See Also

ALLOCATE DESCRIPTOR, SET DESCRIPTOR

D.11.9 OPEN

Name

OPEN -- open a dynamic cursor

Synopsis

```
OPEN cursor_name

OPEN cursor_name USING value [, ... ]

OPEN cursor_name USING SQL DESCRIPTOR descriptor_name
```

Description

OPEN opens a cursor and optionally binds actual values to the placeholders in the cursor's declaration. The cursor must previously have been declared with the DECLARE command. The execution of OPEN causes the query to start executing on the server.

Parameters

cursor_name

The name of the cursor to be opened. This can be an SQL identifier or a host variable.

value

A value to be bound to a placeholder in the cursor. This can be an SQL constant, a host variable, or a host variable with indicator.

descriptor_name

The name of a descriptor containing values to be bound to the placeholders in the cursor. This can be an SQL identifier or a host variable.

Examples

```
EXEC SQL OPEN a END-EXEC.
EXEC SQL OPEN d USING 1, 'test' END-EXEC.
EXEC SQL OPEN c1 USING SQL DESCRIPTOR mydesc END-EXEC.
EXEC SQL OPEN :curname1 END-EXEC.
```

Compatibility

OPEN is specified in the SQL standard.

See Also

DECLARE, CLOSE

Refer to "SQL Commands" in "Reference" in the PostgreSQL Documentation for information on the CLOSE command.

D.11.10 PREPARE

Name

PREPARE -- prepare a statement for execution

Synopsis

```
PREPARE name FROM string
```

Description

PREPARE prepares a statement dynamically specified as a string for execution. This is different from the direct SQL statement PREPARE, which can also be used in embedded programs. The EXECUTE command is used to execute either kind of prepared statement.

Parameters

prepared_name

An identifier for the prepared query.

string

A literal string or a host variable containing a preparable statement, one of the SELECT, INSERT, UPDATE, or DELETE.

Examples

```
MOVE "SELECT * FROM test1 WHERE a = ? AND b = ?" TO ARR OF STMT.
COMPUTE LEN OF STMT = FUNCTION STORED-CHAR-LENGTH (ARR OF STMT).
EXEC SQL ALLOCATE DESCRIPTOR indesc END-EXEC.
EXEC SQL ALLOCATE DESCRIPTOR outdesc END-EXEC.
EXEC SQL PREPARE foo FROM :STMT END-EXEC.

EXEC SQL EXECUTE foo USING SQL DESCRIPTOR indesc INTO SQL DESCRIPTOR outdesc END-EXEC.
```

Compatibility

PREPARE is specified in the SQL standard.

See Also

EXECUTE

Refer to "SQL Commands" in "Reference" in the PostgreSQL Documentation for information on the EXECUTE command.

D.11.11 SET AUTOCOMMIT

Name

SET AUTOCOMMIT -- set the autocommit behavior of the current session

Synopsis

```
SET AUTOCOMMIT { = | TO } { ON | OFF }
```

Description

SET AUTOCOMMIT sets the autocommit behavior of the current database session. By default, embedded SQL programs are not in autocommit mode, so COMMIT needs to be issued explicitly when desired. This command can change the session to autocommit mode, where each individual statement is committed implicitly.

Compatibility

SET AUTOCOMMIT is an extension of PostgreSQL ECOBPG.

D.11.12 SET CONNECTION

Name

SET CONNECTION -- select a database connection

Synopsis

```
SET CONNECTION [ TO | = ] connection_name
```

Description

SET CONNECTION sets the "current" database connection, which is the one that all commands use unless overridden.

Parameters

connection_name

A database connection name established by the CONNECT command.

DEFAULT

Set the connection to the default connection.

Examples

```
EXEC SQL SET CONNECTION TO con2 END-EXEC.
EXEC SQL SET CONNECTION = con1 END-EXEC.
```

Compatibility

SET CONNECTION is specified in the SQL standard.

See Also

CONNECT, DISCONNECT

D.11.13 SET DESCRIPTOR

Name

SET DESCRIPTOR -- set information in an SQL descriptor area

Synopsis

```
SET DESCRIPTOR descriptor_name descriptor_header_item = value [, ... ]

SET DESCRIPTOR descriptor_name VALUE number descriptor_item = value [, ...]
```

Description

SET DESCRIPTOR populates an SQL descriptor area with values. The descriptor area is then typically used to bind parameters in a prepared query execution.

This command has two forms: The first form applies to the descriptor "header", which is independent of a particular datum. The second form assigns values to particular datums, identified by number.

Parameters

descriptor_name

A descriptor name.

descriptor_header_item

A token identifying which header information item to set. Only COUNT, to set the number of descriptor items, is currently supported.

number

The number of the descriptor item to set. The count starts at 1.

descriptor_item

A token identifying which item of information to set in the descriptor. See Section 33.7.1 for a list of supported items.

value

A value to store into the descriptor item. This can be an SQL constant or a host variable.

Examples

```
EXEC SQL SET DESCRIPTOR indesc COUNT = 1 END-EXEC.
EXEC SQL SET DESCRIPTOR indesc VALUE 1 DATA = 2 END-EXEC.
EXEC SQL SET DESCRIPTOR indesc VALUE 1 DATA = :val1 END-EXEC.
EXEC SQL SET DESCRIPTOR indesc VALUE 2 DATA = 'some string', INDICATOR = :val1 END-EXEC.
EXEC SQL SET DESCRIPTOR indesc VALUE 2 INDICATOR = :val2null, DATA = :val2 END-EXEC.
```

Compatibility

SET DESCRIPTOR is specified in the SQL standard.

See Also

ALLOCATE DESCRIPTOR, GET DESCRIPTOR

D.11.14 TYPE

Name

TYPE -- define a new data type

Synopsis

`TYPE` *`type_name`* `IS` *`ctype`*

Description

The TYPE command defines a new COBOL type. It is equivalent to putting a typedef into a declare section.

This command is only recognized when ecobpgpg is run with the -c option.

A level number of 01 is automatically added to type_name item. Thus, the level number must not to be specified externally. To define a group item, a level number needs to be specified to the each subordinate items.

For reasons of internal implementation, "TYPE" must be placed just after "EXEC SQL", without containing newline. For other place, you can use newline.

Parameters

type_name

The name for the new type. It must be a valid COBOL type name.

ctype

A COBOL type specification (including expression format specification).

Examples

```
EXEC SQL TYPE CUSTOMER IS
      02  NAME PIC X(50) VARYING.
      02 PHONE PIC S9(9) COMP. END-EXEC.

EXEC SQL TYPE CUST-IND IS
      02  NAME_IND PIC S9(4) COMP.
      02 PHONE_IND PIC S9(4) COMP. END-EXEC.

EXEC SQL TYPE INTARRAY IS
      02 INT PIC S9(9) OCCURS 20. END-EXEC.
EXEC SQL TYPE STR IS PIC X(50) VARYING. END-EXEC.
EXEC SQL TYPE STRING IS PIC X(10). END-EXEC.
```

Here is an example program that uses EXEC SQL TYPE:

```
EXEC SQL TYPE TT IS
     02 V PIC X(256) VARYING.
     02 I PIC S9(9) COMP. END-EXEC.


EXEC SQL TYPE TT-IND IS
     02 V-IND PIC S9(4) COMP.
     02 I-IND PIC S9(4) COMP. END-EXEC.

EXEC SQL BEGIN DECLARE SECTION END-EXEC.
    01 T TYPE TT.
    01 T-IND TYPE TT-IND.
EXEC SQL END DECLARE SECTION END-EXEC.

    EXEC SQL CONNECT TO testdb AS con1 END-EXEC.

    EXEC SQL SELECT current_database(), 256 INTO :T :T-IND LIMIT 1 END-EXEC.

    DISPLAY "t.v = " ARR OF V OF T.
    DISPLAY "t.i = " I OF T.

    DISPLAY "t_ind.v_ind = " V-IND OF T-IND.
    DISPLAY "t_ind.i_ind = " I-IND OF T-IND.

    EXEC SQL DISCONNECT con1 END-EXEC.
```

Compatibility

The TYPE command is a PostgreSQL extension.

D.11.15 VAR

Name

VAR－ define a variable

Synopsis

```
VAR varname IS ctype
```

Description

The VAR command defines a host variable. It is equivalent to an ordinary COBOL variable definition inside a declare section.

When translating, a level number 01 is added. Thus, the level number must not to be specified externally.

To define a group item, a level number needs to be specified to the each subordinate items.

For reasons of internal implementation, "VAR" must be placed just after "EXEC SQL", without containing newline. For other place, you can use newline.

Parameters

varname

A COBOL variable name.

ctype

A COBOL type specification.

Examples

```
EXEC SQL VAR VC IS PIC X(10) VARYING. END-EXEC.
EXEC SQL VAR BOOL-VAR IS BOOL. END-EXEC.
```

Compatibility

The VAR command is a PostgreSQL extension.

D.11.16 WHENEVER

Name

WHENEVER -- specify the action to be taken when an SQL statement causes a specific class condition to be raised

Synopsis

```
WHENEVER { NOT FOUND | SQLERROR | SQLWARNING } action
```

Description

Define a behavior which is called on the special cases (Rows not found, SQL warnings or errors) in the result of SQL execution.

Parameters

See Section "D.7.1 Setting Callbacks" or a description of the parameters.

Examples

```
EXEC SQL WHENEVER NOT FOUND CONTINUE END-EXEC.
EXEC SQL WHENEVER SQLWARNING SQLPRINT END-EXEC.
EXEC SQL WHENEVER SQLWARNING DO "warn" END-EXEC.
EXEC SQL WHENEVER SQLERROR sqlprint END-EXEC.
EXEC SQL WHENEVER SQLERROR CALL "print2" END-EXEC.
EXEC SQL WHENEVER SQLERROR DO handle_error USING "select" END-EXEC.
EXEC SQL WHENEVER SQLERROR DO sqlnotice USING 0 1 END-EXEC.
EXEC SQL WHENEVER SQLERROR DO "sqlprint" END-EXEC.
EXEC SQL WHENEVER SQLERROR GOTO error_label END-EXEC.
EXEC SQL WHENEVER SQLERROR STOP END-EXEC.
```

A typical application is the use of WHENEVER NOT FOUND GOTO to handle looping through result sets:

```
    EXEC SQL CONNECT TO testdb AS con1 END-EXEC.
    EXEC SQL ALLOCATE DESCRIPTOR d END-EXEC.
    EXEC SQL DECLARE cur CURSOR FOR SELECT current_database(), 'hoge', 256 END-EXEC.
    EXEC SQL OPEN cur END-EXEC.

*   when end of result set reached, break out of while loop
    EXEC SQL WHENEVER NOT FOUND GOTO NOTFOUND END-EXEC.

    PERFORM NO LIMIT
        EXEC SQL FETCH NEXT FROM cur INTO SQL DESCRIPTOR d END-EXEC
        ...
    END-PERFORM.

    NOTFOUND.
    EXEC SQL CLOSE cur END-EXEC.
    EXEC SQL COMMIT END-EXEC.

    EXEC SQL DEALLOCATE DESCRIPTOR d END-EXEC.
    EXEC SQL DISCONNECT ALL END-EXEC.
```

Compatibility

WHENEVER is specified in the SQL standard, but most of the actions are PostgreSQL extensions.

D.12 PostgreSQL Client Applications

This part contains reference information for PostgreSQL client applications and utilities. Not all of these commands are of general utility; some might require special privileges. The common feature of these applications is that they can be run on any host, independent of where the database server resides.

When specified on the command line, user and database names have their case preserved—the presence of spaces or special characters might require quoting. Table names and other identifiers do not have their case preserved, except where documented, and might require quoting.

D.12.1 ecobpg

Name

ecobpg -- embedded SQL COBOL preprocessor

Synopsis

ecobpg [*option*...] *file*...

Description

ecobpg is the embedded SQL preprocessor for COBOL programs. It converts COBOL programs with embedded SQL statements to normal COBOL code by replacing the SQL invocations with special function calls. The output files can then be processed with any COBOL compiler tool chain.

ecobpg will convert each input file given on the command line to the corresponding COBOL output file. Input files preferably have the extension .pco, in which case the extension will be replaced by .cob to determine the output file name. If the extension of the input file is not .pco, then the output file name is computed by appending .cob to the full file name. The output file name can also be overridden using the -o option.

Options

ecobpg accepts the following command-line arguments:

`-c`

Automatically generate certain COBOL code from SQL code. Currently, this works for EXEC SQL TYPE.

-I directory

Specify an additional include path, used to find files included via EXEC SQL INCLUDE. Defaults are: (current directory), /usr/local/include, the PostgreSQL include directory which is defined at compile time (default: /usr/local/pgsql/include), and /usr/include, in that order.

-o filename

Specifies that ecobpg should write all its output to the given filename.

-f format

Specifies the COBOL code notation. For "format", specify either of the following. If omitted, "fixed" is used.

fixed

Specifies fixed format notation. Up to 72 columns can be specified for area B. Characters in column 73 and beyond are deleted in the precompiled source.

variable

Specifies variable format notation. Up to 251 columns can be specified for area B. Characters in column 252 and beyond are deleted in the precompiled source.

-r option

Selects run-time behavior. Option can be one of the following:

prepare

Prepare all statements before using them. Libecpg will keep a cache of prepared statements and reuse a statement if it gets executed again. If the cache runs full, libecpg will free the least used statement.

questionmarks

Allow question mark as placeholder for compatibility reasons. This used to be the default long ago.

-t

Turn on autocommit of transactions. In this mode, each SQL command is automatically committed unless it is inside an explicit transaction block. In the default mode, commands are committed only when EXEC SQL COMMIT is issued.

--varchar-with-named-member

When converting VARCHAR host variable, adding name of the variable to members as prefix. Instead of LEN and ARR, (varname)-ARR and (varname)-LEN will be used.

--bytea-with-named-member

When converting bytea host variable, adding name of the variable to members as prefix. Instead of LEN and ARR, (varname)-ARR and (varname)-LEN will be used.

-E encode

Specify the COBOL source encoding: "UTF8", "SJIS", or "EUC_JP".

If this option is omitted, the encoding is processed based on the locale.

-v

Print additional information including the version and the "include" path.

--version

Print the ecobpg version and exit.

-?
--help

Show help about ecobpg command line arguments, and exit.

Notes

When compiling the preprocessed COBOL code files, the compiler needs to be able to find the library files in the PostgreSQL include directory.

Programs using COBOL code with embedded SQL have to be linked against the libecpg library, for example using the linker options.

The value of either of these directories that is appropriate for the installation can be found out using pg_config.

See

Refer to "pg_config" in "Reference" in the PostgreSQL Documentation.

Examples

If you have an embedded SQL COBOL source file named prog1.pco, you can create an executable program using the following command:

```
ecobpg prog1.pco
```

Appendix E Quantitative Limits

This appendix lists the quantitative limits of Fujitsu Enterprise Postgres.

Table E.1 Length of identifier

Item	Limit
Database name	Up to 63 bytes (*1) (*2)
Schema name	Up to 63 bytes (*1) (*2)
Table name	Up to 63 bytes (*1) (*2)
View name	Up to 63 bytes (*1) (*2)
Index name	Up to 63 bytes (*1) (*2)
Table space name	Up to 63 bytes (*1) (*2)
Cursor name	Up to 63 bytes (*1) (*2)
Function name	Up to 63 bytes (*1) (*2)
Aggregate function name	Up to 63 bytes (*1) (*2)
Trigger name	Up to 63 bytes (*1) (*2)
Constraint name	Up to 63 bytes (*1) (*2)
Conversion name	Up to 63 bytes (*1) (*2)
Role name	Up to 63 bytes (*1) (*2)
Cast name	Up to 63 bytes (*1) (*2)
Collation sequence name	Up to 63 bytes (*1) (*2)
Encoding method conversion name	Up to 63 bytes (*1) (*2)
Domain name	Up to 63 bytes (*1) (*2)
Extension name	Up to 63 bytes (*1) (*2)
Operator name	Up to 63 bytes (*1) (*2)
Operator class name	Up to 63 bytes (*1) (*2)
Operator family name	Up to 63 bytes (*1) (*2)
Rewrite rule name	Up to 63 bytes (*1) (*2)
Sequence name	Up to 63 bytes (*1) (*2)
Text search settings name	Up to 63 bytes (*1) (*2)
Text search dictionary name	Up to 63 bytes (*1) (*2)
Text search parser name	Up to 63 bytes (*1) (*2)
Text search template name	Up to 63 bytes (*1) (*2)
Data type name	Up to 63 bytes (*1) (*2)
Enumerator type label	Up to 63 bytes (*1) (*2)

*1: This is the character string byte length when converted by the server character set character code.

*2: If an identifier that exceeds 63 bytes in length is specified, the excess characters are truncated and it is processed.

Table E.2 Database object

Item	Limit
Number of databases	Less than 4,294,967,296 (*1)

Item	Limit
Number of schemas	Less than 4,294,967,296 (*1)
Number of tables	Less than 4,294,967,296 (*1)
Number of views	Less than 4,294,967,296 (*1)
Number of indexes	Less than 4,294,967,296 (*1)
Number of table spaces	Less than 4,294,967,296 (*1)
Number of functions	Less than 4,294,967,296 (*1)
Number of aggregate functions	Less than 4,294,967,296 (*1)
Number of triggers	Less than 4,294,967,296 (*1)
Number of constraints	Less than 4,294,967,296 (*1)
Number of conversion	Less than 4,294,967,296 (*1)
Number of roles	Less than 4,294,967,296 (*1)
Number of casts	Less than 4,294,967,296 (*1)
Number of collation sequences	Less than 4,294,967,296 (*1)
Number of encoding method conversions	Less than 4,294,967,296 (*1)
Number of domains	Less than 4,294,967,296 (*1)
Number of extensions	Less than 4,294,967,296 (*1)
Number of operators	Less than 4,294,967,296 (*1)
Number of operator classes	Less than 4,294,967,296 (*1)
Number of operator families	Less than 4,294,967,296 (*1)
Number of rewrite rules	Less than 4,294,967,296 (*1)
Number of sequences	Less than 4,294,967,296 (*1)
Number of text search settings	Less than 4,294,967,296 (*1)
Number of text search dictionaries	Less than 4,294,967,296 (*1)
Number of text search parsers	Less than 4,294,967,296 (*1)
Number of text search templates	Less than 4,294,967,296 (*1)
Number of data types	Less than 4,294,967,296 (*1)
Number of enumerator type labels	Less than 4,294,967,296 (*1)
Number of default access privileges defined in the ALTER DEFAULT PRIVILEGES statement	Less than 4,294,967,296 (*1)
Number of large objects	Less than 4,294,967,296 (*1)
Number of index access methods	Less than 4,294,967,296 (*1)

*1: The total number of all database objects must be less than 4,294,967,296.

Table E.3 Schema element

Item	Limit
Number of columns that can be defined in one table	From 250 to 1600 (according to the data type)
Table row length	Up to 400 gigabytes
Number of columns comprising a unique constraint	Up to 32 columns
Data length comprising a unique constraint	Less than 2,000 bytes (*1) (*2)

Item	Limit
Table size	Up to one terabyte
Search condition character string length in a trigger definition statement	Up to 800 megabytes (*1) (*2)
Item size	Up to 1 gigabyte

*1: Operation might proceed correctly even if operations are performed with a quantity outside the limits.

*2: This is the character string byte length when converted by the server character set character code.

Table E.4 Index

Item	Limit
Number of columns comprising a key (including VCI)	Up to 32 columns
Key length (other than VCI)	Less than 2,000 bytes (*1)

*1: This is the character string byte length when converted by the server character set character code.

Table E.5 Data types and attributes that can be handled

Item			Limit
Character	Data length		Data types and attributes that can be handled (*1)
	Specification length (*n*)		Up to 10,485,760 characters (*1)
Numeric	External decimal expression		Up to 131,072 digits before the decimal point, and up to 16,383 digits after the decimal point
	Internal binary expression	2 bytes	From -32,768 to 32,767
		4 bytes	From -2,147,483,648 to 2,147,483,647
		8 bytes	From -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
	Internal decimal expression		Up to 13,1072 digits before the decimal point, and up to 16,383 digits after the decimal point
	Floating point expression	4 bytes	From -3.4E+38 to -7.1E-46, 0, or from 7.1E-46 to 3.4E+38
		8 bytes	From -1.7E+308 to -2.5E-324, 0, or from 2.5E-324 to 1.7E+308
bytea			Up to one gigabyte minus 53 bytes
Large object			Up to two gigabytes

*1: This is the character string byte length when converted by the server character set character code.

Table E.6 Function definition

Item	Limit
Number of arguments that can be specified	Up to 100
Number of variable names that can be specified in the declarations section	No limit

Item	Limit
Number of SQL statements or control statements that can be specified in a function processing implementation	No limit

Table E.7 Data operation statement

Item	Limit
Maximum number of connections for one process in an application (remote access)	4,000 connections
Number of expressions that can be specified in a selection list	Up to 1,664
Number of tables that can be specified in a FROM clause	No limit
Number of unique expressions that can be specified in a selection list/DISTINCT clause/ORDER BY clause/GROUP BY clause within one SELECT statement	Up to 1,664
Number of expressions that can be specified in a GROUP BY clause	No limit
Number of expressions that can be specified in an ORDER BY clause	No limit
Number of SELECT statements that can be specified in a UNION clause/INTERSECT clause/EXCEPT clause	Up to 4,000 (*1)
Number of nestings in joined tables that can be specified in one view	Up to 4,000 (*1)
Number of functions or operator expressions that can be specified in one expression	Up to 4,000 (*1)
Number of expressions that can be specified in one row constructor	Up to 1,664
Number of expressions that can be specified in an UPDATE statement SET clause	Up to 1,664
Number of expressions that can be specified in one row of a VALUES list	Up to 1,664
Number of expressions that can be specified in a RETURNING clause	Up to 1,664
Total expression length that can be specified in the argument list of one function specification	Up to 800 megabytes (*2)
Number of cursors that can be processed simultaneously by one session	No limit
Character string length of one SQL statement	Up to 800 megabytes (*1) (*3)
Number of input parameter specifications that can be specified in one dynamic SQL statement	No limit
Number of tokens that can be specified in one SQL statement	Up to 10,000
Number of values that can be specified as a list in a WHERE clause IN syntax	No limit
Number of expressions that can be specified in a USING clause	No limit
Number of JOINs that can be specified in a joined table	Up to 4,000 (*1)
Number of expressions that can be specified in COALESCE	No limit
Number of WHEN clauses that can be specified for CASE in a simple format or a searched format	No limit

Item	Limit
Data size per record that can be updated or inserted by one SQL statement	Up to one gigabyte minus 53 bytes
Number of objects that can share a lock simultaneously	Up to 256,000 (*1)

*1: Operation might proceed correctly even if operations are performed with a quantity outside the limits.

*2: The total number of all database objects must be less than 4,294,967,296.

*3: This is the character string byte length when converted by the server character set character code.

Table E.8 Data sizes

Item	Limit
Data size per record for input data files (COPY statement, psql command \copy meta command)	Up to 800 megabytes (*1)
Data size per record for output data files (COPY statement, psql command \copy meta command)	Up to 800 megabytes (*1)

*1: Operation might proceed correctly even if operations are performed with a quantity outside the limits.

Appendix F Reference

F.1 JDBC Driver

See

Refer to the Java API Reference for information on PostgreSQL JDBC driver.

F.2 ODBC Driver

F.2.1 List of Supported APIs

The following table shows the support status of APIs:

Function name	Support status
SQLAllocConnect	Y
SQLAllocEnv	Y
SQLAllocHandle	Y
SQLAllocStmt	Y
SQLBindCol	Y
SQLBindParameter	Y
SQLBindParam	Y
SQLBrowseConnect	Y
SQLBulkOperations	Y
SQLCancel	Y
SQLCancelHandle	N
SQLCloseCursor	Y
SQLColAttribute	Y
SQLColAttributeW	Y
SQLColAttributes	Y
SQLColAttributesW	Y
SQLColumnPrivileges	Y
SQLColumnPrivilegesW	Y
SQLColumns	Y
SQLColumnsW	Y
SQLCompleteAsync	N
SQLConnect	Y
SQLConnectW	Y
SQLCopyDesc	Y
SQLDataSources	Y
SQLDataSourcesW	Y
SQLDescribeCol	Y

Function name	Support status
SQLDescribeColW	Y
SQLDescribeParam	Y
SQLDisconnect	Y
SQLDriverConnect	Y
SQLDriverConnectW	Y
SQLDrivers	Y
SQLEndTran	Y
SQLError	Y
SQLErrorW	Y
SQLExecDirect	Y
SQLExecDirectW	Y
SQLExecute	Y
SQLExtendedFetch	Y
SQLFetch	Y
SQLFetchScroll	Y
SQLForeignKeys	Y
SQLForeignKeysW	Y
SQLFreeConnect	Y
SQLFreeEnv	Y
SQLFreeHandle	Y
SQLFreeStmt	Y
SQLGetConnectAttr	Y
SQLGetConnectAttrW	Y
SQLGetConnectOption	Y
SQLGetConnectOptionW	Y
SQLGetCursorName	Y
SQLGetCursorNameW	Y
SQLGetData	Y
SQLGetDescField	Y
SQLGetDescFieldW	Y
SQLGetDescRec	Y
SQLGetDescRecW	Y
SQLGetDiagField	Y
SQLGetDiagFieldW	Y
SQLGetDiagRec	Y
SQLGetDiagRecW	Y
SQLGetEnvAttr	Y
SQLGetFunctions	Y
SQLGetInfo	Y

Function name	Support status
SQLGetInfoW	Y
SQLGetStmtAttr	Y
SQLGetStmtAttrW	Y
SQLGetStmtOption	Y
SQLGetTypeInfo	Y
SQLGetTypeInfoW	Y
SQLMoreResults	Y
SQLNativeSql	Y
SQLNativeSqlW	Y
SQLNumParams	Y
SQLNumResultCols	Y
SQLParamData	Y
SQLParamOptions	Y
SQLPrepare	Y
SQLPrepareW	Y
SQLPrimaryKeys	Y
SQLPrimaryKeysW	Y
SQLProcedureColumns	Y
SQLProcedureColumnsW	Y
SQLProcedures	Y
SQLProceduresW	Y
SQLPutData	Y
SQLRowCount	Y
SQLSetConnectAttr	Y
SQLSetConnectAttrW	Y
SQLSetConnectOption	Y
SQLSetConnectOptionW	Y
SQLSetCursorName	Y
SQLSetCursorNameW	Y
SQLSetDescField	Y
SQLSetDescRec	Y
SQLSetEnvAttr	Y
SQLSetParam	Y
SQLSetPos	Y
SQLSetScrollOptions	N
SQLSetStmtAttr	Y
SQLSetStmtAttrW	Y
SQLSetStmtOption	Y
SQLSpecialColumns	Y

Function name	Support status
SQLSpecialColumnsW	Y
SQLStatistics	Y
SQLStatisticsW	Y
SQLTablePrivileges	Y
SQLTablePrivilegesW	Y
SQLTables	Y
SQLTablesW	Y
SQLTransact	Y

Y: Supported
N: Not supported

F.3 .NET Data Provider

There are the following ways to develop applications using Fujitsu Npgsql .NET Data Provider:

- Use the Fujitsu Npgsql.NET Data Provider API (classes and methods) directly.

  Fujitsu Npgsql .NET Data Provider is created based on the open source software Npgsql. Refer to the "Npgsql - .Net Data Provider for Postgresql" for information on the APIs:

  See

  Refer to the Installation and Setup Guide for Client for the version of Npgsql that Fujitsu Npgsql .NET Data Provider is based on.

- Use the API of the .NET System.Data.Common namespace

  It is possible to create applications that do not rely on a provider when you use the System.Data.Common namespace. Refer to the "Writing Provider-Independent Code in ADO.NET" in MSDN Library for information.

See

Refer to the Npgsql API Reference for information on Npgsql API classes and methods.

F.4 C Library (libpq)

See

Refer to "libpq - C Library" in "Client Interfaces" in the PostgreSQL Documentation.

F.5 Embedded SQL in C

See

Refer to "ECPG - Embedded SQL in C" in "Client Interfaces" in the PostgreSQL Documentation.

Index

[A]

Additional Notes on each Type Plugin....42

[B]

BIND_VARIABLE....88

[C]

CLOSE_CURSOR....88
Code examples for applications....55,65
COLUMN_VALUE....88
Comparison operator....3

[D]

DBMS_SQL....86
DECODE....81
DEFINE_COLUMN....89
DUAL Table....81

[E]

Encoding System Settings....24,48
Entry information of subprogram....68
Example of specifying the hint clause....56,67
EXECUTE....89

[F]

FETCH_ROWS....90

[L]

Language settings....7,23,48
Libraries to use....68

[N]

Notes on automatically generating update-type SQL statements 45
Notes on metadata....45
Notes on the Query Builder....44
NVL....84

[O]

OPEN_CURSOR....90
Outer Join Operator (+)....78

[P]

PARSE....90
Path of library....68
Path of the library file....68
Pattern matching....3
Precompiling example....56,67

[S]

Scan Using a Vertical Clustered Index (VCI)....113
Settings for encrypting communication data for connection to the server....7
String functions and operators....3
SUBSTR....83

[T]

Type Plugins....41

[U]

Use the API of the .NET System.Data.Common namespace.208
Use the Fujitsu Npgsql.NET Data Provider API (classes and methods) directly....208

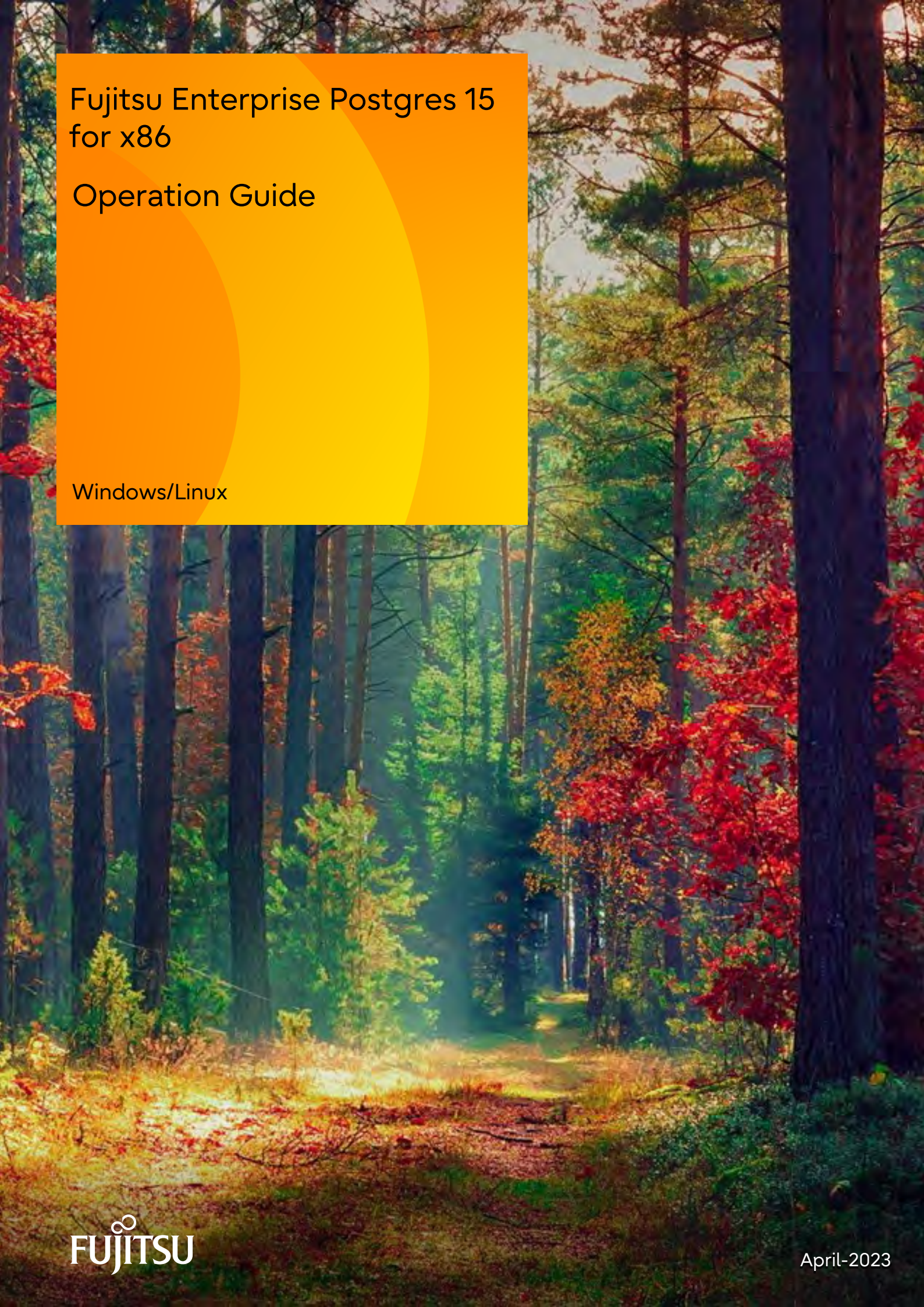
[W]

When setting from outside with environment variables....49
When specifying in the connection URI....49
When using Add-OdbcDsn....19
When using ODBCConf.exe....18

Fujitsu Enterprise Postgres 15 for x86

Operation Guide

Windows/Linux



FUJITSU

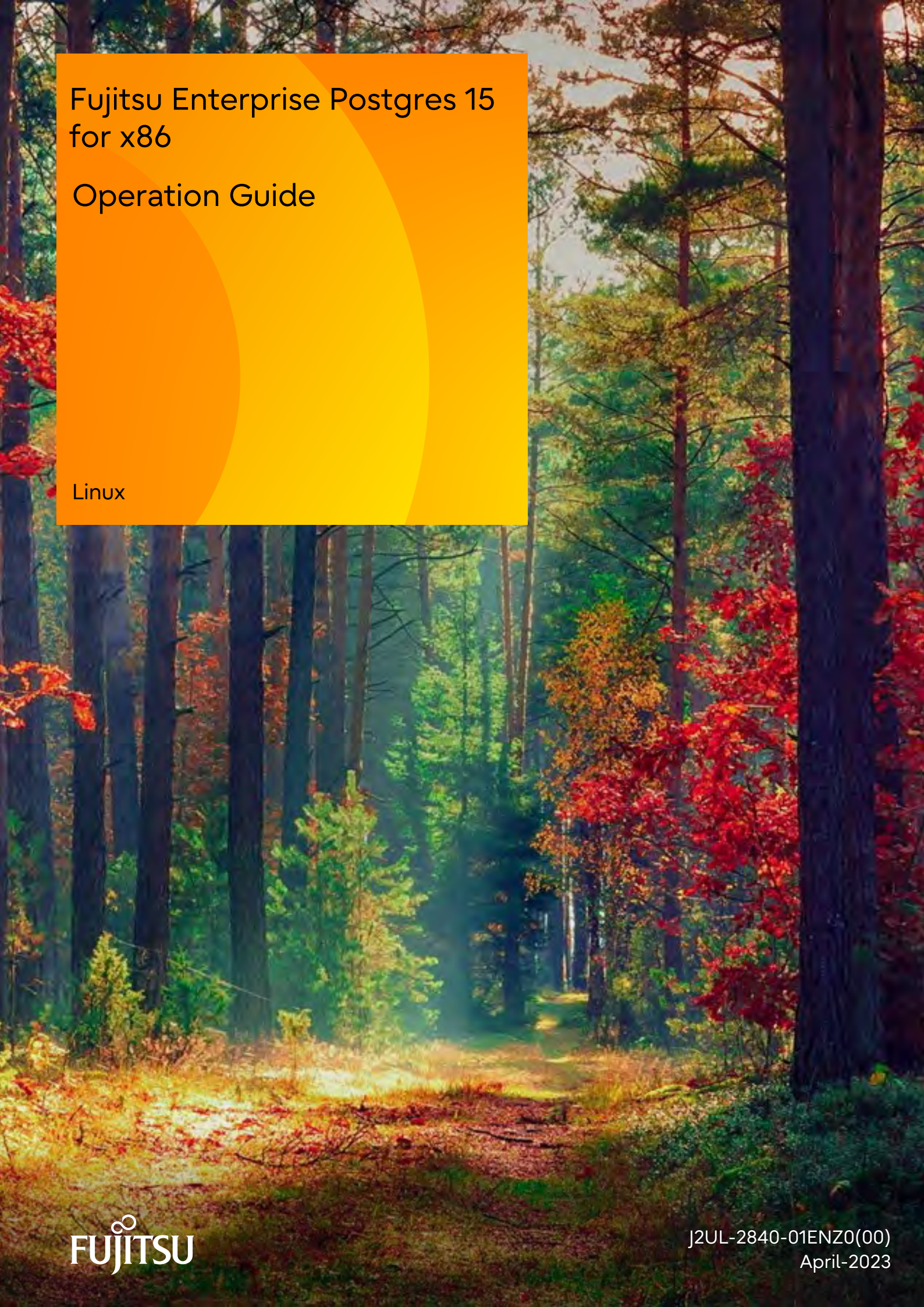
April-2023

Linux >
Windows >

Fujitsu Enterprise Postgres 15 for x86

Operation Guide

Linux



FUJITSU

J2UL-2840-01ENZ0(00)
April-2023

Preface

Purpose of this document

The Fujitsu Enterprise Postgres database system extends the PostgreSQL features and runs on the Linux platform.

This document is the Fujitsu Enterprise Postgres Operation Guide.

Intended readers

This document is intended for those who install and operate Fujitsu Enterprise Postgres.

Readers of this document are assumed to have general knowledge of:

- PostgreSQL
- SQL
- Linux

Structure of this document

This document is structured as follows:

Chapter 1 Operating Fujitsu Enterprise Postgres

Describes how to operate Fujitsu Enterprise Postgres.

Chapter 2 Starting an Instance and Creating a Database

Describes how to start a Fujitsu Enterprise Postgres instance, and how to create a database.

Chapter 3 Backing Up the Database

Describes how to back up the database.

Chapter 4 Configuring Secure Communication Using Secure Sockets Layer

Describes communication data encryption between the client and the server.

Chapter 5 Protecting Storage Data Using Transparent Data Encryption

Describes how to encrypt the data to be stored in the database.

Chapter 6 Using Transparent Data Encryption with Key Management Systems as Keystores

Describes the operation of transparent data encryption when a key management system is used as a keystore.

Chapter 7 Data Masking

Describes the data masking feature.

Chapter 8 Periodic Operations

Describes the periodic database operations that must be performed on Fujitsu Enterprise Postgres.

Chapter 9 Streaming Replication Using WebAdmin

Describes how to create a streaming replication cluster using WebAdmin.

Chapter 10 Installing and Operating the In-memory Feature

Describes how to install and operate the in-memory feature.

Chapter 11 Parallel Query

Describes the factors taken into consideration by Fujitsu Enterprise Postgres when performing parallel queries.

Chapter 12 High-Speed Data Load

Describes how to install and operate high-speed data load.

Chapter 13 Global Meta Cache

Describes how to use Grobal Meta Cache feature.

Chapter 14 Local Meta Cache Limit

Describes how to use Local Meta Cache Limit feature.

Chapter 15 Backup/Recovery Using the Copy Command

Describes backup and recovery using the copy command created by the user.

Chapter 16 Actions when an Error Occurs

Describes how to perform recovery when disk failure or data corruption occurs.

Appendix A Parameters

Describes the Fujitsu Enterprise Postgres parameters.

Appendix B System Administration Functions

Describes the system administration functions of Fujitsu Enterprise Postgres.

Appendix C System Views

Describes how to use the system view in Fujitsu Enterprise Postgres.

Appendix D Tables Used by Transparent Data Encryption

Describes the tables used by the transparent data encryption feature.

Appendix E Tables Used by Data Masking

Describes the tables used by the data masking feature.

Appendix F Tables Used by High-Speed Data Load

Describes the tables used by high-speed data load.

Appendix G Starting and Stopping the Web Server Feature of WebAdmin

Describes how to start and stop WebAdmin (Web server feature).

Appendix H WebAdmin Wallet

Describes how to use the Wallet feature of WebAdmin.

Appendix I WebAdmin Disallow User Inputs Containing Hazardous Characters

Describes characters not allowed in WebAdmin.

Appendix J Copy Command Samples that Use the Advanced Copy Feature of the ETERNUS Disk Array

Describes copy command samples that use the advanced copy feature of the ETERNUS disk array.

Appendix K Collecting Failure Investigation Data

Describes how to collect information for initial investigation.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version


```
Edition 1.0: April  2023
```


Copyright

Copyright 2015-2023 Fujitsu Limited

Contents

Chapter 1 Operating Fujitsu Enterprise Postgres....1
1.1 Operating Methods....1
1.2 Starting WebAdmin....2
1.2.1 Logging in to WebAdmin....2
1.3 Starting pgAdmin....3
1.3.1 Starting pgAdmin....3
1.3.2 Adding an Instance....4
1.3.3 Connecting/Disconnecting an Instance....6
1.3.4 Changing the pgAdmin Display Language....6
1.4 Operations Using Commands....6
1.5 Operating Environment of Fujitsu Enterprise Postgres....7
1.5.1 Operating Environment....7
1.5.2 File Composition....9
1.6 Notes on Compatibility of Applications Used for Operations....10
1.7 Notes on Upgrading Database Instances....10
1.7.1 Additional Steps for upgrading to Fujitsu Enterprise Postgres with Vertical Clustered Index (VCI) Enabled....11

Chapter 2 Starting an Instance and Creating a Database....13
2.1 Starting and Stopping an Instance....13
2.1.1 Using WebAdmin....13
2.1.2 Using Server Commands....15
2.2 Creating a Database....16
2.2.1 Using pgAdmin....16
2.2.2 Using Client Commands....17

Chapter 3 Backing Up the Database....18
3.1 Periodic Backup....19
3.2 Backup Methods....19
3.2.1 Using WebAdmin....19
3.2.2 Using Server Commands....20

Chapter 4 Configuring Secure Communication Using Secure Sockets Layer....24
4.1 Configuring Communication Data Encryption....24
4.1.1 Issuing a Certificate....25
4.1.2 Deploying a Server Certificate File and a Server Private Key File....25
4.1.3 Distributing a CA Certificate File to the Client....25
4.1.4 Configuring the Operating Environment for the Database Server....25
4.1.5 Configuring the Operating Environment for the Client....25
4.1.6 Performing Database Multiplexing....26

Chapter 5 Protecting Storage Data Using Transparent Data Encryption....27
5.1 Protecting Data Using Encryption....27
5.2 Setting the Master Encryption Key....28
5.3 Opening the Keystore....29
5.4 Encrypting a Tablespace....29
5.5 Checking an Encrypted Tablespace....30
5.6 Managing the Keystore....31
5.6.1 Changing the Master Encryption Key....31
5.6.2 Changing the Keystore Passphrase....31
5.6.3 Enabling Automatic Opening of the Keystore....31
5.6.4 Backing Up and Recovering the Keystore....32
5.7 Backing Up and Restoring/Recovering the Database....34
5.8 Importing and Exporting the Database....36
5.9 Encrypting Existing Data....36
5.10 Operations in Cluster Systems....37
5.10.1 HA Clusters that do not Use Database Multiplexing....37

5.10.2 Database Multiplexing Mode....37
5.11 Security-Related Notes....38
5.12 Tips for Installing Built Applications....39

Chapter 6 Using Transparent Data Encryption with Key Management Systems as Keystores....40
6.1 Protecting Data Using Encryption....40
6.2 Setting the Master Encryption Key....41
6.3 Opening the Keystore....42
6.4 Encrypting a Tablespace....42
6.5 Checking an Encrypted Tablespace....42
6.6 Managing the Keystore....42
6.6.1 Changing the Master Encryption Key....42
6.6.2 Enabling Automatic Opening of the Keystore....42
6.6.3 Changing Credentials for Key Management Systems....43
6.6.4 Verifying the Master Encryption Key....43
6.6.5 Changes to the Key Management System....43
6.7 Backing Up and Restoring/Recovering the Database....44
6.8 Importing and Exporting the Database....45
6.9 Encrypting Existing Data....45
6.10 Operations in Cluster Systems....45
6.10.1 HA Clusters that do not Use Database Multiplexing....45
6.10.2 Database Multiplexing Mode....46
6.11 Security-Related Notes....46
6.12 Tips for Installing Built Applications....47

Chapter 7 Data Masking....48
7.1 Masking Policy....48
7.1.1 Masking Target....49
7.1.2 Masking Type....49
7.1.3 Masking Condition....49
7.1.4 Masking Format....50
7.2 Usage Method....52
7.2.1 Creating a Masking Policy....53
7.2.2 Changing a Masking Policy....54
7.2.3 Confirming a Masking Policy....54
7.2.4 Enabling and Disabling a Masking Policy....55
7.2.5 Deleting a Masking Policy....56
7.3 Data Types for Masking....56
7.4 Security Notes....57

Chapter 8 Periodic Operations....58
8.1 Configuring and Monitoring the Log....58
8.2 Monitoring Disk Usage and Securing Free Space....58
8.2.1 Monitoring Disk Usage....58
8.2.2 Securing Free Disk Space....58
8.3 Automatically Closing Connections....59
8.4 Monitoring the Connection State of an Application....59
8.4.1 Using the View (pg_stat_activity)....60
8.4.2 Using pgAdmin....61
8.5 Reorganizing Indexes....61
8.6 Monitoring Database Activity....62
8.6.1 Information that can be Collected....63
8.6.2 Collection Configuration....64
8.6.3 Information Reset....64

Chapter 9 Streaming Replication Using WebAdmin....66
9.1 Creating a Standby Instance....66
9.2 Promoting a Standby Instance....68

9.3 Converting an Asynchronous Replication to Synchronous ... 68
9.4 Converting a Synchronous Replication to Asynchronous ... 69
9.5 Joining a Replication Cluster ... 69

Chapter 10 Installing and Operating the In-memory Feature ... 71
10.1 Installing Vertical Clustered Index (VCI) ... 71
10.1.1 Evaluating whether to Install VCI ... 71
10.1.2 Estimating Resources ... 72
10.1.3 Setting up ... 72
10.1.3.1 Setting Parameters ... 72
10.1.3.2 Installing the Extensions ... 73
10.1.3.3 Creating a VCI ... 73
10.1.3.4 Confirming that the VCI has been Created ... 74
10.1.4 Data that can Use VCI ... 74
10.1.4.1 Relation Types ... 74
10.1.4.2 Data Types ... 75
10.2 Operating VCI ... 76
10.2.1 Commands that cannot be Used for VCI ... 77
10.2.2 Data Preload Feature ... 78

Chapter 11 Parallel Query ... 79
11.1 CPU Load Calculation ... 79
11.2 Increase of Workers during Runtime ... 79
11.3 Statistics View Displays the Action State ... 79

Chapter 12 High-Speed Data Load ... 81
12.1 Installing High-Speed Data Load ... 81
12.1.1 Deciding whether to Install ... 81
12.1.2 Estimating Resources ... 81
12.1.3 Setup ... 82
12.1.3.1 Setting Parameters ... 82
12.1.3.2 Installing the Extension ... 83
12.2 Using High-Speed Data Load ... 83
12.2.1 Loading Data ... 83
12.2.2 Checking Progress ... 84
12.2.3 Recovering from a Data Load that Ended Abnormally ... 85
12.3 Removing High-Speed Data Load ... 86
12.3.1 Removing the Extension ... 86

Chapter 13 Global Meta Cache ... 88
13.1 Usage ... 88
13.1.1 Deciding Whether to Enable the Global Meta Cache Feature ... 88
13.1.2 Estimating Memory for Global Meta Cache ... 88
13.1.3 How the GMC Memory Area Is Used ... 88
13.1.4 Enabling the Global Meta Cache Feature ... 88
13.1.5 Estimating Resources ... 89
13.2 Statistics ... 89
13.2.1 System View ... 89

Chapter 14 Local Meta Cache Limit ... 90
14.1 Usage ... 90
14.1.1 Deciding Whether to Enable the Local Meta Cache Limit Feature ... 90
14.1.2 How to Set Parameters for the Local Meta Cache Limit Feature ... 90
14.1.3 Cache Removal when Local Meta Cache Limit is Enabled ... 90
14.1.4 Performance Impact and Parameter Tuning of the Local Meta Cache Limit Feature ... 91

Chapter 15 Backup/Recovery Using the Copy Command ... 93
15.1 Configuration of the Copy Command ... 93
15.2 Backup Using the Copy Command ... 97

15.3 Recovery Using the Copy Command ... 98
15.4 Copy Command Interface ... 99
15.4.1 Copy Command for Backup ... 99
15.4.2 Copy Command for Recovery ... 101

Chapter 16 Actions when an Error Occurs ... 103
16.1 Recovering from Disk Failure (Hardware) ... 104
16.1.1 Using WebAdmin ... 104
16.1.2 Using Server Command ... 105
16.2 Recovering from Data Corruption ... 109
16.2.1 Using WebAdmin ... 110
16.2.2 Using the pgx_rcvall Command ... 110
16.3 Recovering from an Incorrect User Operation ... 111
16.3.1 Using WebAdmin ... 111
16.3.2 Using the pgx_rcvall Command ... 112
16.4 Actions in Response to an Application Error ... 113
16.4.1 When using the view (pg_stat_activity) ... 114
16.4.2 Using the ps Command ... 114
16.4.3 Using pgAdmin ... 115
16.5 Actions in Response to an Access Error ... 116
16.6 Actions in Response to Insufficient Space on the Data Storage Destination ... 116
16.6.1 Using a Tablespace ... 116
16.6.2 Replacing the Disk with a Larger Capacity Disk ... 116
16.6.2.1 Using WebAdmin ... 117
16.6.2.2 Using Server Commands ... 117
16.7 Actions in Response to Insufficient Space on the Backup Data Storage Destination ... 119
16.7.1 Temporarily Saving Backup Data ... 119
16.7.1.1 Using WebAdmin ... 119
16.7.1.2 Using Server Commands ... 120
16.7.2 Replacing the Disk with a Larger Capacity Disk ... 122
16.7.2.1 Using WebAdmin ... 123
16.7.2.2 Using Server Commands ... 123
16.8 Actions in Response to Insufficient Space on the Transaction Log Storage Destination ... 127
16.8.1 Replacing the Disk with a Larger Capacity Disk ... 127
16.8.1.1 Using WebAdmin ... 127
16.8.1.2 Using Server Commands ... 128
16.9 Errors in More Than One Storage Disk ... 130
16.10 Actions in Response to Instance Startup Failure ... 130
16.10.1 Errors in the Configuration File ... 130
16.10.2 Errors Caused by Power Failure or Mounting Issues ... 130
16.10.3 Other Errors ... 130
16.10.3.1 Using WebAdmin ... 131
16.10.3.2 Using Server Commands ... 131
16.11 Actions in Response to Failure to Stop an Instance ... 131
16.11.1 Using WebAdmin ... 131
16.11.2 Using Server Commands ... 131
16.11.2.1 Stopping the Instance Using the Fast Mode ... 132
16.11.2.2 Stopping the Instance Using the Immediate Mode ... 132
16.11.2.3 Forcibly Stopping the Server Process ... 132
16.12 Actions in Response to Failure to Create a Streaming Replication Standby Instance ... 133
16.13 Actions in Response to Error in a Distributed Transaction ... 133
16.14 I/O Errors Other than Disk Failure ... 134
16.14.1 Network Error with an External Disk ... 134
16.14.2 Errors Caused by Power Failure or Mounting Issues ... 135
16.15 Anomaly Detection and Resolution ... 135
16.15.1 Port Number and Backup Storage Path Anomalies ... 135
16.15.2 Mirroring Controller Anomalies ... 136

Appendix A Parameters....137

Appendix B System Administration Functions....146
B.1 WAL Mirroring Control Functions....146
B.2 Transparent Data Encryption Control Functions....146
B.2.1 pgx_open_keystore....146
B.2.2 pgx_set_master_key....147
B.2.3 pgx_declare_external_master_key....147
B.2.4 pgx_set_keystore_passphrase....148
B.3 Data Masking Control Functions....148
B.3.1 pgx_alter_confidential_policy....148
B.3.2 pgx_create_confidential_policy....154
B.3.3 pgx_drop_confidential_policy....157
B.3.4 pgx_enable_confidential_policy....158
B.3.5 pgx_update_confidential_values....160
B.4 VCI Data Load Control Function....161
B.5 High-Speed Data Load Control Functions....161

Appendix C System Views....162
C.1 pgx_tablespaces....162
C.2 pgx_stat_lwlock....162
C.3 pgx_stat_latch....162
C.4 pgx_stat_walwriter....163
C.5 pgx_stat_sql....163
C.6 pgx_stat_gmc....164
C.7 pgx_stat_progress_loader....164

Appendix D Tables Used by Transparent Data Encryption....165
D.1 pgx_tde_master_key....165

Appendix E Tables Used by Data Masking....166
E.1 pgx_confidential_columns....166
E.2 pgx_confidential_policies....166
E.3 pgx_confidential_values....167

Appendix F Tables Used by High-Speed Data Load....168
F.1 pgx_loader_state....168

Appendix G Starting and Stopping the Web Server Feature of WebAdmin....169
G.1 Starting the Web Server Feature of WebAdmin....169
G.2 Stopping the Web Server Feature of WebAdmin....169

Appendix H WebAdmin Wallet....171
H.1 Creating a Credential....171
H.2 Using a Credential....172

Appendix I WebAdmin Disallow User Inputs Containing Hazardous Characters....173

Appendix J Copy Command Samples that Use the Advanced Copy Feature of the ETERNUS Disk Array....174

Appendix K Collecting Failure Investigation Data....176

Index....177

Chapter 1 Operating Fujitsu Enterprise Postgres

This chapter describes how to operate Fujitsu Enterprise Postgres.

1.1 Operating Methods

There are two methods of managing Fujitsu Enterprise Postgres operations:

- Operation management using GUI tools
- Operation management using commands

See

Before performing database multiplexing using database multiplexing, refer to "Database Multiplexing Mode" in the Cluster Operation Guide (Database Multiplexing).

Operation management using GUI tools

This involves managing operations using the WebAdmin and pgAdmin GUI tools.

- Management using WebAdmin

  This removes the requirement for complex environment settings and operational design for backup and recovery that is usually required for running a database. It enables you to easily and reliably monitor the state of the database, create a streaming replication cluster, back up the database, and restore it even if you do not have expert knowledge of databases.

- Management using pgAdmin

  When developing applications and maintaining the database, you can use pgAdmin to perform simple operations on database objects, such as:

  - Rebuild indexes and update statistics
  - Create, delete, and update database objects

  In addition, from pgAdmin of Fujitsu Enterprise Postgres, you can use the expanded features provided by Fujitsu Enterprise Postgres on the PostgreSQL SQL commands.

  See

  Refer to pgAdmin Help for information on the expanded features of pgAdmin provided by Fujitsu Enterprise Postgres.

Operation management using commands

You can use commands for configuring and operating the database and managing operations.

Note

- You cannot combine WebAdmin and server commands to perform the following operations:
  - Use commands to operate an instance created using WebAdmin.
  - Use WebAdmin to recover a database backed up using commands.

  For instances created with WebAdmin, however, backup can be obtained with the pgx_dmpall command. Also, WebAdmin can perform recovery by using the backup obtained with the pgx_dmpall command.

- To operate an instance created using the initdb command in WebAdmin, the instance needs to be imported using WebAdmin.

- You can perform backup and restoration in pgAdmin, but the backup data obtained with WebAdmin and pgx_dmpall is not compatible with the backup data obtained with pgAdmin.
- Refer to pgAdmin Help for other notes on pgAdmin.

Features used in each phase

The following table lists the features used in each phase for GUI-based operations and command-based operations.

Operation		Operation with the GUI	Operation with commands
Setup	Creating an instance	WebAdmin is used. The server machine capacity, and the optimum parameter for operations using WebAdmin, are set automatically.	The configuration file is edited directly using the initdb command.
	Creating a standby instance	WebAdmin is used. WebAdmin performs a base backup of the source instance and creates a standby instance.	A standby instance is created using the pg_basebackup command.
	Changing the configuration files	WebAdmin is used.	The configuration file is edited directly.
Starting and stopping an instance		WebAdmin is used.	The pg_ctl command is used.
Creating a database		This is defined using pgAdmin of the GUI tool, or using the psql command or the application after specifying the DDL statement.	
Backing up the database		WebAdmin, or the pgx_dmpall command, is used.	It is recommended that the pgx_dmpall command be used. Recovery to the latest database can be performed.
Database recovery		WebAdmin is used.	To use the backup that was performed using the pgx_dmpall command, the pgx_rcvall command is used.
Monitoring	Database errors	The status in the WebAdmin window can be checked. (*1)	The messages that are output to the database server log are monitored (*1)
	Disk space	The status in the WebAdmin window can be checked. A warning will be displayed if the free space falls below 20%. (*1)	This is monitored using the df command of the operating system, for example. (*1)
	Connection status	This can be checked using pgAdmin of the GUI tool, or referencing pg_stat_activity of the standard statistics view from psql or the application.	

*1: This can be used together with system log monitoring using operations management middleware (Systemwalker Centric Manager, for example).

1.2 Starting WebAdmin

This section describes how to start and log in to WebAdmin.

1.2.1 Logging in to WebAdmin

This section describes how to log in to WebAdmin.

User environment

It is recommended to use the following browsers with WebAdmin:

- Microsoft Edge (Build41 or later)

WebAdmin will work with other browsers, such as Firefox and Chrome, however, the look and feel may be slightly different.

Startup URL for WebAdmin

In the browser address bar, type the startup URL of the WebAdmin window in the following format:

```
http://hostNameOrIpAddress:portNumber/
```

- *hostNameOrIpAddress*: The host name or IP address of the server where WebAdmin is installed.
- *portNumber*: The port number of WebAdmin. The default port number is 27515.

Example

For a server with IP address "192.0.2.0" and port number "27515"

```
http://192.0.2.0:27515/
```

Display the startup windows. From this window you can log in to WebAdmin or access the product documentation.

Note

- You must start the Web server feature of WebAdmin before using WebAdmin.
- Refer to "Appendix G Starting and Stopping the Web Server Feature of WebAdmin" for information on how to start the Web server feature of WebAdmin.

Log in to WebAdmin

Click [Launch WebAdmin] in the startup URL window to start WebAdmin and display the login window.

To log in, specify the following values:

- [User name]: User name (OS user account) of the instance administrator
- [Password]: Password corresponding to the user name

Point

Use the OS user account as the user name of the instance administrator. Refer to "Creating an Instance Administrator" in the Installation and Setup Guide for Server for details.

1.3 Starting pgAdmin

This section describes how to start pgAdmin, how to add an instance required for managing a database, and how to connect to and disconnect from the instance.

You can use pgAdmin on the Windows client.

1.3.1 Starting pgAdmin

This section explains how to start pgAdmin if you are using it from the product "Fujitsu Enterprise Postgres Client (64bit) *x* SP*z*" (where "*x*" is the product version, and "*z*" is the product level (*x* SPz)).

Click [Start] >> [All apps] >> [Fujitsu Enterprise Postgres Client(64bit) *x* SP*z*] and start [pgAdmin 4 (64bit) (*x* SP*z*)].

Note

- You must start the instance to be connected to before using pgAdmin.
- Refer to "2.1 Starting and Stopping an Instance" for information on how to start an instance.
- When using pgAdmin4 with Microsoft Edge, enable network access by loopback in Microsoft Edge. Also, add Microsoft Edge to the loopback exclusion list.

1.3.2 Adding an Instance

This section describes how to add an instance to be connected to.

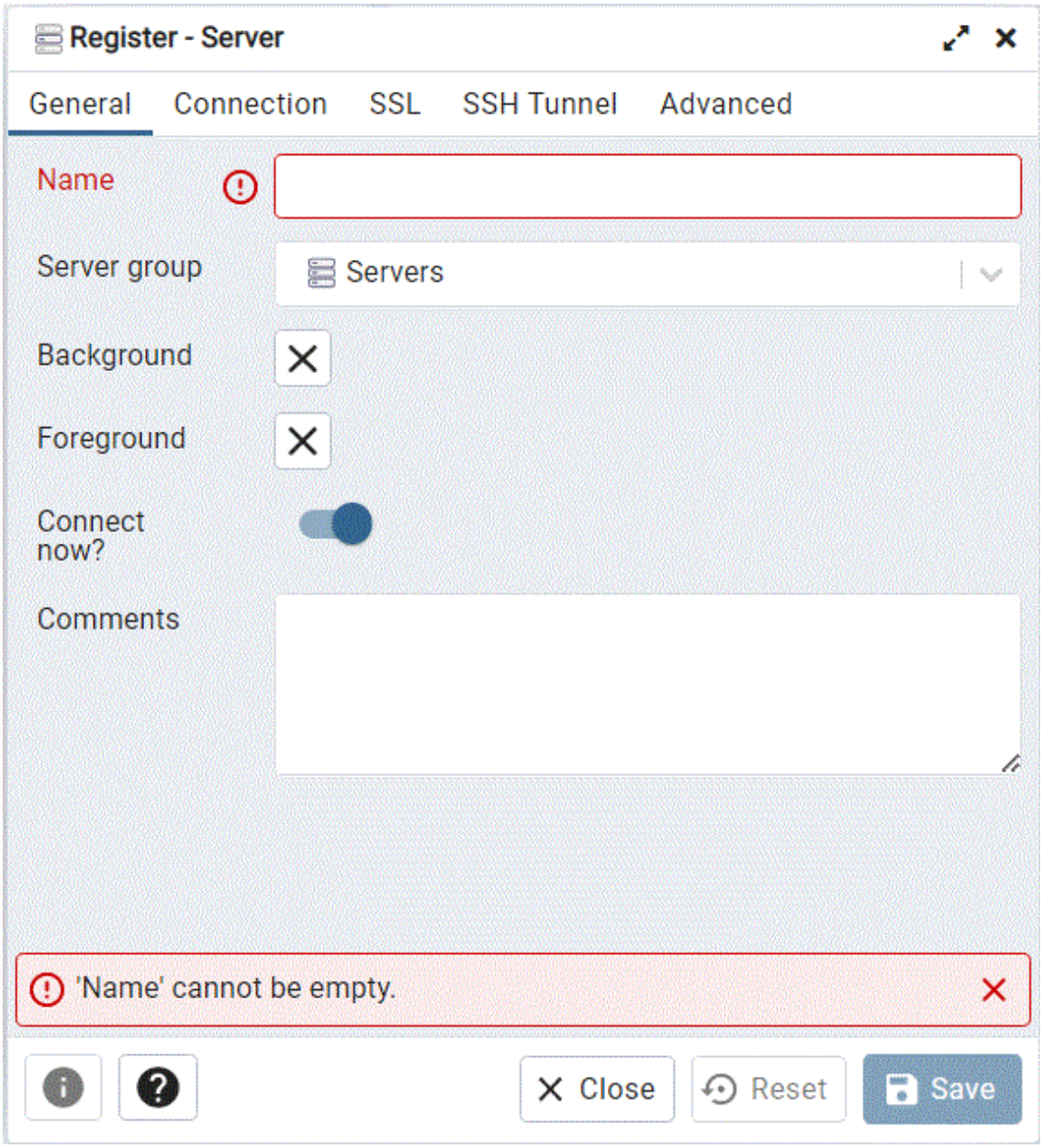
Note

If you use a link-local address with version 6 of the TCP/IP protocol, you may encounter the following error when registering the server with pgAdmin. Therefore, do not use link-local addresses.

```
unsupported format character '"' (0x22) at index 96
```

1. In the [Browser] pane, right-click [Servers], and then click [Register] >> [Server...].

2. In the [Register - Server] window, specify a value for each item.



[General] tab

- [Name]: Name of the instance to be managed

[Connection] tab

- [Host name/address]: Host name or IP address of the server where Fujitsu Enterprise Postgres is installed
- [Port]: Port number of the instance
- [Maintenance database]: Database name to which the client will connect
- [Username]: User name of the instance administrator
- [Password]: Password for the user name specified in [Username]

Note

If you select [Save password], the Fujitsu Enterprise Postgres connection password is stored in the following location. Set the appropriate access permissions for the password file to protect it from unauthorized access.

- %APPDATA%\Roaming\pgAdmin\pgadmin4.db

1.3.3 Connecting/Disconnecting an Instance

This section describes how to connect pgAdmin to an instance, and how to disconnect it.

Note

To connect to an instance created using WebAdmin, you must first configure the settings in the [Client authentication] window of WebAdmin to permit connection from pgAdmin.

See

Refer to "Changing the settings" in the Installation and Setup Guide for Server for information on the [Client authentication] window of WebAdmin.

Connecting to an instance

Starting pgAdmin does not connect it to any instance.

To connect to an instance, in the [Browser] pane, right-click the server, and then click [Connect Server].

If a password was not saved when the instance was added, enter a password in the password entry window that is displayed.

Disconnecting from an instance

To disconnect from an instance, in the [Browser] pane, right-click the server, and then click [Disconnect from Server].

1.3.4 Changing the pgAdmin Display Language

This section describes the procedure for changing the display language of pgAdmin.

1. Start pgAdmin.
2. Click [File] >> [Preferences].
3. Click [Miscellaneous] >> [User language] in the left pane of the [Preferences] dialog.
4. Select the language you want to display from the [User language] drop-down list displayed in the right pane, and click the [Save] button.
5. Refresh page by clicking the [Refresh] button.

1.4 Operations Using Commands

You can operate and manage the database using the following commands:

- Server commands

  This group of commands includes commands for creating a database cluster and controlling the database. You can run these commands on the server where the database is operating.

  To use these commands, you must configure the environment variables.

See

- Refer to "PostgreSQL Server Applications" under "Reference" in the PostgreSQL Documentation, or "Reference" for information on server commands.
- Refer to "Configure the environment variables" in the procedure to create instances in "Using the initdb Command" in the Installation and Setup Guide for Server for information on configuring the environment variables.

- Client commands

  This group of commands includes the psql command and commands for extracting the database cluster to a script file. These commands can be executed on the client that can connect to the database, or on the server on which the database is running.

  To use these commands, you must configure the environment variables.

See

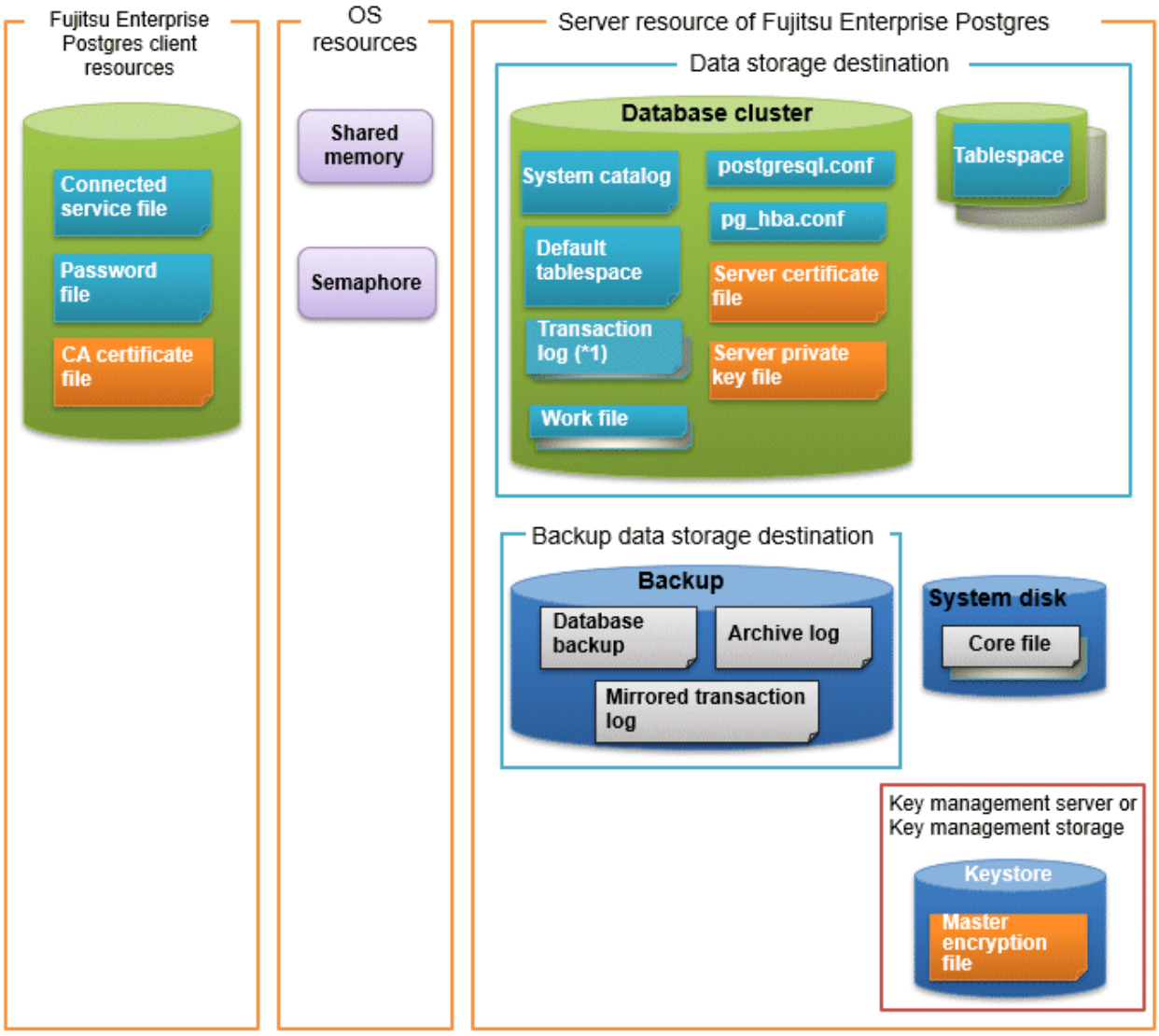
- Refer to "PostgreSQL Client Applications" under "Reference" in the PostgreSQL Documentation, or "Reference" for information on client commands.
- Refer to "Configuring Environment Variables" in the Installation and Setup Guide for Client for information on the values to be set in the environment variables.

1.5 Operating Environment of Fujitsu Enterprise Postgres

This section describes the operating environment and the file composition of Fujitsu Enterprise Postgres.

1.5.1 Operating Environment

The following figure shows the configuration of the Fujitsu Enterprise Postgres operating environment. The tables given below list the roles of the OS resources and Fujitsu Enterprise Postgres resources.



*1: To distribute the I/O load, place the transaction log on a different disk from the data storage destination.

Table 1.1 OS resources

Type	Role
Shared memory	Used when a database process exchanges information with an external process.
Semaphore	

Table 1.2 Fujitsu Enterprise Postgres client resources

Type	Role
Connection service file	Specifies information, such as the host name, user name, and password, for connecting to Fujitsu Enterprise Postgres.
Password file	Securely manages the password for connecting to Fujitsu Enterprise Postgres.
CA certificate file	CA (certificate authority) certificate used for server authentication when encrypting communication data.

Table 1.3 Server resources of Fujitsu Enterprise Postgres

Type	Role
Database cluster	Database storage area on the database storage disk. It is a collection of databases managed by an instance.
System catalog	Contains information required for the system to run, including the database definition information and the operation information created by the user.
Default tablespace	Contains table files and index files stored by default.
Transaction log	Contains log information in case of a crash recovery or rollback. This is the same as the WAL (Write Ahead Log).
Work file	Work file used when executing applications or commands.
postgresql.conf	Contains information that defines the operating environment of Fujitsu Enterprise Postgres.
pg_hba.conf	Fujitsu Enterprise Postgres uses this file to authenticate individual client hosts.
Server certificate file	Contains information about the server certificate to be used when encrypting communication data and authenticating a server.
Server private key file	Contains information about the server private key to be used when encrypting communication data and authenticating a server
Tablespace	Stores table files and index files in a separate area from the database cluster. Specify a space other than that under the database cluster.
Backup	Stores the data required for recovering the database when an error, such as disk failure, occurs.
Database backup	Contains the backup data for the database.
Archive log	Contains the log information for recovery.
Mirrored transaction log (mirrored WAL)	Enables a database cluster to be restored to the state immediately before an error even if both the database cluster and transaction log fail when performing backup/recovery operations using the pgx_dmpall command or WebAdmin.
Core file	Fujitsu Enterprise Postgres process core file that is output when an error occurs during a Fujitsu Enterprise Postgres process.
Key management server or key management storage	Server or storage where the master encryption key file is located.
Master encryption key file	Contains the master encryption key to be used when encrypting storage data. The master encryption key file is managed on the key management server or key management storage.

1.5.2 File Composition

Fujitsu Enterprise Postgres consists of the following files for controlling and storing the database. The table below shows the relationship between the number of such files and their location within a single instance.

Table 1.4 Number of files within a single instance and how to specify their location

File type	Required	Quantity	How to specify the location
Program files	Y	Multiple	Note that "<*x*>" indicates the product version. /opt/fsepv<*x*>server64
Database cluster	Y	1	Specify using WebAdmin or server commands.
Tablespace	Y	Multiple	Specify a space other than that under the database cluster, using pgAdmin or the DDL statement.

File type	Required	Quantity	How to specify the location
Backup	Y	Multiple	Specify using WebAdmin or server commands.
Core file	Y	Multiple	Specify using WebAdmin, server commands, or postgresql.conf.
Server certificate file (*1)	N	1	Specify using postgresql.conf.
Server private key file (*1)	N	1	Specify using postgresql.conf.
Master encryption key file (*1)	N	1	Specify the directory created as the key store using postgresql.conf.
Connection service file (*1)	N	1	Specify using environment variables.
Password file (*1)	N	1	Specify using environment variables.
CA certificate file (*1)	N	1	Specify using environment variables.

Y: Mandatory
N: Optional

*1: Set manually when using the applicable feature.

Note

- Do not place files for use with Fujitsu Enterprise Postgres in a directory mounted over the network except when creating a database space in a storage device on a network.
  Examples include NFS (Network File System) and CIFS (Common Internet File System).
  This is because the database might hang if the network fails.
- If anti-virus software is used, set scan exception settings for directories so that none of the files that comprise Fujitsu Enterprise Postgres are scanned for viruses. Alternatively, if the files that comprise Fujitsu Enterprise Postgres are to be scanned for viruses, stop Fujitsu Enterprise Postgres and perform the scan when tasks that use Fujitsu Enterprise Postgres are not operating.

1.6 Notes on Compatibility of Applications Used for Operations

When you upgrade Fujitsu Enterprise Postgres to a newer version, there may be some effect on applications due to improvements or enhancements in functionality.

Take this into account when creating applications so that you can maintain compatibility after upgrading to a newer version of Fujitsu Enterprise Postgres.

See

Refer to "Notes on Application Compatibility" in the Application Development Guide for details.

1.7 Notes on Upgrading Database Instances

When upgrading FUJITSU Enterprise Postgres 9.4 or newer database instances to FUJITSU Enterprise Postgres 10 or later using pg_upgrade, there are certain steps you need to follow.

Before using pg_upgrade, remove the following extensions from all databases in the instance, except "template0":

- pg_stat_statements
- pgx_io

- pgx_paging
- pgx_network
- pgx_network_err
- pgx_cpu
- pgx_memory
- pgx_swap
- pgx_disk
- pgx_process
- pgx_log
- oracle_compatible
- pg_dbms_stats
- pg_hint_plan

For all databases except "template0", execute the following command to remove these extensions:

```
DROP EXTENSION extensionName;
```

Once the pg_upgrade operation is complete, for all databases except "template0", execute the following command to re-create these extensions as required:

```
CREATE EXTENSION extensionName;
```

Note

- It is strongly recommended to back up the database using pg_dump before performing pg_upgrade or using DROP EXTENSION.
- If there are any columns created in the user tables using a data type from these extensions, then DROP EXTENSION will also drop these columns. Therefore, it is essential that alternate upgrade mechanisms are considered instead of pg_upgrade, in such scenarios. These may include pg_dump/pg_restore.

1.7.1 Additional Steps for upgrading to Fujitsu Enterprise Postgres with Vertical Clustered Index (VCI) Enabled

When upgrading FUJITSU Enterprise Postgres 11.0 or earlier instances that are using the VCI extension to FUJITSU Enterprise Postgres 12 or later using pg_upgrade, additional steps must be performed because of the incompatibility of the VCI extension between FUJITSU Enterprise Postgres 12 or later and FUJITSU Enterprise Postgres 11 or earlier.

Follow the procedure below in all databases in the FUJITSU Enterprise Postgres 11 or earlier instance, except "template0".

Before upgrading

1. Obtain the CREATE INDEX Definitions

   Run the query below to list all the VCI indexes created in the database. Ensure that these indexes are re-created in the FUJITSU Enterprise Postgres 12 or later instance after pg_upgrade has finished.

   ```
   SELECT nspname || '.' || relname AS index_relname,* FROM pg_class, pg_namespace
     WHERE relnamespace = pg_namespace.oid AND relam IN (SELECT oid FROM pg_am WHERE amname='vci');
   ```

   For each index_relname listed above, execute the commands below to obtain the CREATE INDEX definition (to use the same SQL syntax while re-creating the indexes on the FUJITSU Enterprise Postgres 12 or later instance).

   ```
   SELECT pg_get_indexdef('indexName'::regclass);
   ```

2. Drop the VCI indexes and VCI extension along with all its dependencies.

   To remove all the VCI indexes and VCI internal objects that are created in Fujitsu Enterprise Postgres, execute the commands below. VCI internal objects will be created in FUJITSU Enterprise Postgres 12 or later automatically when CREATE EXTENSION for VCI is executed.

   ```
   DROP EXTENSION VCI CASCADE;
   ```

   Note

   To restore the VCI extension in the FUJITSU Enterprise Postgres 11 or earlier instance, execute CREATE EXTENSION.

After upgrading

Once the pg_upgrade operation is complete, for all databases except "template0", execute CREATE EXTENSION to create the VCI extension, and then execute CREATE INDEX for all the VCI indexes as required.

Chapter 2 Starting an Instance and Creating a Database

This chapter describes basic operations, from starting an instance to creating a database.

2.1 Starting and Stopping an Instance

This section describes how to start and stop an instance.

- 2.1.1 Using WebAdmin
- 2.1.2 Using Server Commands

Point

To automatically start or stop an instance when the operating system on the database server is started or stopped, refer to "Configuring Automatic Start and Stop of an Instance" in the Installation and Setup Guide for Server and configure the settings.

Note

The collected statistics are initialized if an instance is stopped in the "Immediate" mode or if it is abnormally terminated. To prepare for such initialization of statistics, consider regular collection of the statistics by using the SELECT statement. Refer to "The Statistics Collector" in "Server Administration" in the PostgreSQL Documentation for information on the statistics.

2.1.1 Using WebAdmin

WebAdmin enables you to start or stop an instance and check its operating status.

Starting an instance

Start an instance by using the [Instances] tab in WebAdmin.

is displayed when an instance is stopped.

To start a stopped instance, click .

Stopping an instance

Stop an instance by using the [Instances] tab in WebAdmin.

 is displayed when an instance is active.

To stop an active instance, click .

Stop mode

Select the mode in which to stop the instance. The following describes the operations of the modes:

Stop mode	Connected clients	Backup being executed using the command
Smart mode (*1)	Waits for all connected clients to be disconnected.	Waits for backups being executed using the command to finish.
Fast mode	Rolls back all transactions being executed and forcibly disconnects clients.	Terminates backups being executed using the command.
Immediate mode	All server processes are terminated immediately. Crash recovery is executed the next time the instance is started.	

Stop mode	Connected clients	Backup being executed using the command
Kill process mode	Send SIGKILL to the process and abort all active transactions. This will lead to a crash-recovery run at the next restart.	

*1: When the processing to stop the instance in the Smart mode has started and you want to stop immediately, use the following procedure:

1. Restart the Web server feature of WebAdmin.
2. In the [Instances] tab, click .
3. In the [Instances] tab, click , and select the Immediate mode to stop the instance.

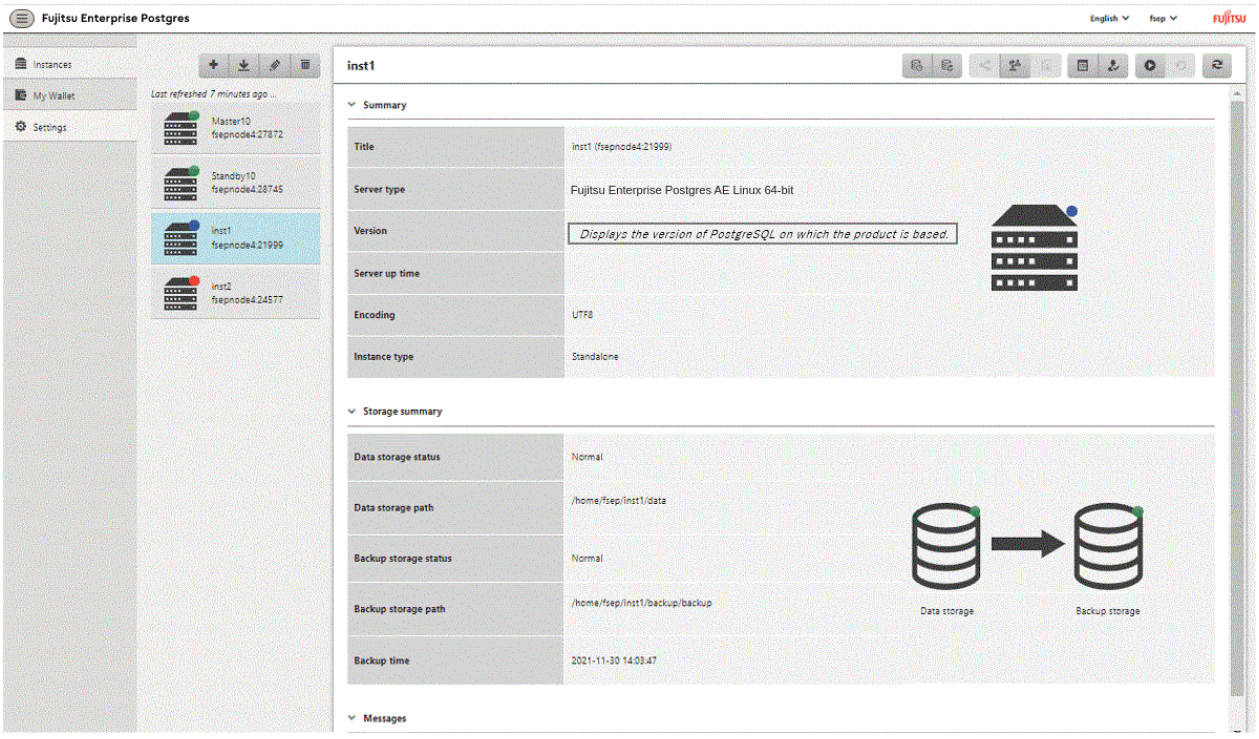
Checking the operating status of an instance

You can check the operating status of an instance by using the [Instances] tab. The following indicators are used to show the status of a resource.

Status indicator	Explanation
	The resource is operating normally.
	The resource is stopped.
	There is an error in the resource.
	An operation is in progress on this resource or the status is being checked.
	The resource is not operating optimally and needs intervention.

If an instance stops abnormally, remove the cause of the stoppage and start the instance by using WebAdmin.

Figure 2.1 Example of operating status indicators



Note

- When operating WebAdmin, click to update the status. WebAdmin will reflect the latest status of the operation or the instance resources from the server.
- If an error occurs while communicating with the server, there may be no response from WebAdmin. When this happens, close the browser and then log in again. If this does not resolve the issue, check the system log of the server and confirm whether a communication error has occurred.
- The following message is output during startup of an instance when the startup process is operating normally, therefore, the user does not need to be aware of this message:

```
FATAL:  the database system is starting up
```

2.1.2 Using Server Commands

Server commands enable you to start or stop an instance and check its operating status.

To use sever commands, configure the environment variables.

See

Refer to " Configure the environment variables" in the procedure to create instances in " Using the initdb Command" in the Installation and Setup Guide for Server for information on configuring the environment variables.

Starting an instance

Use the pg_ctl command to start an instance.

Specify the following values in the pg_ctl command:

- Specify "start" as the mode.
- Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

If an application, command, or process tries to connect to the database while the instance is starting up, the message "FATAL:the database system is starting up(11189)" is output. However, this message may also be output if the instance is started without the -W option specified. This message is output by the pg_ctl command to check if the instance has started successfully. Therefore, ignore this message if there are no other applications, commands, or processes that connect to the database.

Example

```
> pg_ctl start -D /database/inst1
```

Note

If the -W option is specified, the command will return without waiting for the instance to start. Therefore, it may be unclear as to whether the instance startup was successful or failed.

Stopping an instance

Use the pg_ctl command to stop an instance.

Specify the following values in the pg_ctl command:

- Specify "stop" as the mode.

- Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

Example

```
> pg_ctl stop -D /database/inst1
```

Checking the operating status of an instance

Use the pg_ctl command to check the operating status of an instance.

Specify the following values in the pg_ctl command:

- Specify "status" as the mode.
- Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

Example

When the instance is active:

```
> pg_ctl status -D /database/inst1
pg_ctl: server is running (PID: 1234)
```

When the instance is inactive:

```
> pg_ctl status -D /database/inst1
pg_ctl: no server running.
```

See

Refer to "pg_ctl" under "Reference" in the PostgreSQL Documentation for information on pg_ctl command.

2.2 Creating a Database

This section explains how to create a database.

- 2.2.1 Using pgAdmin
- 2.2.2 Using Client Commands

2.2.1 Using pgAdmin

Follow the procedure below to define a database using pgAdmin.

1. In the pgAdmin window, right-click [Databases] in the [Browser] pane, and then click [Create] >> [Database...] to display a [Create - Database] window.
2. Specify appropriate values for the following items in the [Create - Database] window.
   - [General] tab
     - [Database]: Name of the database to be managed
3. Click [Save] to create the database.

2.2.2 Using Client Commands

Follow the procedure below to define a database using client commands.

An example of operations on the server is shown below.

1. Use psql command to connect to the postgres database.
   Execute psql postgres.

```
> psql postgres
psql (<x>) (*1)
Type "help" for help.
```

   *1: <*x*> indicates the PostgreSQL version on which this product is based.

2. Create the database.
   To create the database, execute the CREATE DATABASE databaseName; statement.

```
postgres=# CREATE DATABASE db01;
CREATE DATABASE
```

3. Confirm that the database is created.
   Execute \l+, and confirm that the name of the database created in step 2 is displayed.

```
postgres=# \l+
```

4. Disconnect from the postgres database.
   Execute \q to terminate the psql command.

```
postgres=# \q
```

You can create a database using the createdb command.

See

Refer to "Creating a Database" in "Tutorial" in the PostgreSQL Documentation for information on creating a database using the createdb command.

Chapter 3 Backing Up the Database

This chapter describes how to back up the database.

Backup methods

The following backup methods enable you to recover data to a backup point or to the state immediately preceding disk physical breakdown or data logical failure.

- Backup using WebAdmin

  This method enables you to back up data through intuitive window operations using the GUI.

  WebAdmin is used for recovery.

- Backup using the pgx_dmpall command

  Execute the pgx_dmpall command with a script to perform automatic backup.

  To back up data automatically, you must register the process in the automation software of the operating system. Follow the procedure given in the documentation for your operating system.

  The pgx_rcvall command is used for recovery.

Information

By using a copy command created by the user, the pgx_dmpall command and the pgx_rcvall command can back up database clusters and tablespaces to any destination and recover them from any destination using any copy method. Refer to "Chapter 15 Backup/Recovery Using the Copy Command" for details.

Approximate backup time

The formula for deriving the approximate backup time when you use WebAdmin or the pgx_dmpall command is as follows:

```
backupTime = dataStorageDestinationUsage / diskWritePerformance x 1.5
```

- *dataStorageDestinationUsage*: Disk usage at the data storage destination
- *diskWritePerformance*: Maximum data volume (bytes/second) that can be written per second in the system environment where operation is performed
- 1.5: Coefficient to factor in tasks other than disk write (which is the most time-consuming step)

If using the copy command with the pgx_dmpall command, the backup time will depend on the implementation of the copy command.

Note

- Backup operation cannot be performed on an instance that is part of a streaming replication cluster in standby mode.
- Use the selected backup method continuously.

  There are several differences, such as the data format, across the backup methods. For this reason, the following restrictions apply:

  - It is not possible to use one method for backup and another for recovery.
  - It is not possible to convert one type of backup data to a different type of backup data.
- Mirrored WALs can be used only for backup/recovery using the pgx_dmpall command or WebAdmin.
- There are several considerations for the backup of the keystore and backup of the database in case the data stored in the database is encrypted. Refer to the following for details:
  - 5.6.4 Backing Up and Recovering the Keystore
  - 5.7 Backing Up and Restoring/Recovering the Database

- If you have defined a tablespace, back it up. If you do not back it up, directories for the tablespace are not created during recovery, which may cause the recovery to fail. If the recovery fails, refer to the system log, create the tablespace, and then perform the recovery process again.

Information

The following methods can also be used to perform backup. Performing a backup using these methods allows you to restore to the point when the backup was performed.

- Backup using an SQL-based dump

  Dump the data by using SQL. This backup method also enables data migration.

- File system level backup

  This backup method requires you to stop the instance and use OS commands to backup database resources as files.

- Backup by continuous archiving

  This is the standard backup method for PostgreSQL.

Refer to "Backup and Restore" in "Server Administration" in the PostgreSQL Documentation for information on these backup methods.

3.1 Periodic Backup

It is recommended that you perform backup periodically.

Backing up data periodically using WebAdmin or the pgx_dmpall command has the following advantages:

- This method reduces disk usage, because obsolete archive logs (transaction logs copied to the backup data storage destination) are deleted. It also minimizes the recovery time when an error occurs.

Backup cycle

The time interval when backup is performed periodically is called the backup cycle. For example, if backup is performed every morning, the backup cycle is 1 day.
The backup cycle depends on the jobs being run, but on Fujitsu Enterprise Postgres it is recommended that operations are run with a backup cycle of at least once per day.

3.2 Backup Methods

This section describes the methods for backing up the database.

- 3.2.1 Using WebAdmin
- 3.2.2 Using Server Commands

3.2.1 Using WebAdmin

You can use WebAdmin to perform backup and check the backup status.

Note

- If backup is disabled for an instance, you will not be able to back up or restore the instance. Refer to "[Backup]" in "Creating an Instance" in the Installation and Setup Guide for Server for details.
- If the data to be stored in the database is to be encrypted, it is necessary to enable the automatic opening of the keystore before doing so. Refer to "5.6.3 Enabling Automatic Opening of the Keystore" for details.

- WebAdmin uses the labels "Data storage path", "Backup storage path" and "Transaction log path" to indicate "data storage destination", "backup data storage destination" and "transaction log storage destination" respectively. In this manual these terms are used interchangeably.

Backup operation

Follow the procedure below to back up the database.

1. Select the database to back up

   In the [Instances] tab, select the instance to be backed up and click .

2. Back up the database

   The [Backup] dialog box is displayed. To perform backup, click [Yes].
   An instance is automatically started when backup is performed.

Backup status

If an error occurs and backup fails, [Error] is displayed adjacent to [Data storage status] or [Backup storage status] in the [Instances] tab. An error message is also displayed in the message list.

In this case, the backup data is not optimized. Ensure that you check the backup result whenever you perform backup. If backup fails, [Solution] appears to the right of the error message. Clicking this button displays information explaining how to resolve the cause of the error. Remove the cause of failure, and perform backup again.

3.2.2 Using Server Commands

Use the pgx_dmpall command and pgx_rcvall command to perform backup and check the backup result.

Preparing for backup

You must prepare for backup before actually starting the backup process.

Follow the procedure below.

See

Refer to "Preparing Directories to Deploy Resources" in the Installation and Setup Guide for Server for information on the location of directories required for backup and for points to take into account.

1. Prepare the backup data storage disk

   For backup, prepare a separate disk unit from the database storage disk and mount it using the operating system commands.

2. Create a directory where the backup data will be stored

   Create an empty directory.

   Set appropriate permissions so that only the instance administrator can access the directory.

   Example

   ```
   # mkdir /backup/inst1
   # chown fsepuser:fsepuser /backup/inst1
   # chmod 700 /backup/inst1
   ```

3. Specify the settings required for backup

   Stop the instance, and set the following parameters in the postgresql.conf file.

   Start the instance after editing the postgresql.conf file.

Parameter name	Setting	Description
backup_destination	Name of the directory where the backup data will be stored	Specify the name of the directory where the backup data will be stored. Appropriate privileges that allow only the instance administrator to access the directory must already be set. Place the backup data storage destination directory outside the data storage destination directory, the tablespace directory, and the transaction log storage destination directory.
archive_mode	on	Specify the archive log mode. Specify [on] (execute).
archive_command	'*installationDirectory*/bin/pgx_walcopy.cmd "%p" "*backupDataStorageDestinationDirectory*/archived_wal/%f"'	Specify the path name of the command that will save the transaction log and the storage destination.
archive_library	"(default)	Specify the archive library. Specify ["](default).

Refer to "Appendix A Parameters" and "Write Ahead Log" under "Server Administration" in the PostgreSQL Documentation for information on the parameters.

Backup operation (file backup)

Use the pgx_dmpall command to perform file backup. You can even embed the pgx_dmpall command in OS automation software to perform backup.

The backup data is stored in the directory specified in the backup_destination parameter of postgresql.conf.

Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

Example

```
> pgx_dmpall -D /database/inst1
```

Note

Backup stores the data obtained during the backup and the backup data of the data obtained during previous backup.

If the data to be stored in the database is encrypted, refer to the following and back up the keystore:

- 5.6.4 Backing Up and Recovering the Keystore

Backup status

Use the pgx_rcvall command to check the backup status.

Specify the following values in the pgx_rcvall command:

- The -l option indicates backup data information.

- Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

```
> pgx_rcvall -l -D /database/inst1
Date                    Status          Dir
2022-03-01 13:30:40     COMPLETE        /backup/inst1/2022-03-01_13-30-40
```

If an error occurs and backup fails, a message is output to the system log.

In this case, the backup data is not optimized. Ensure that you check the backup result whenever you perform backup. If backup fails, remove the cause of failure and perform backup again.

See

Refer to "pgx_dmpall" and "pgx_rcvall" in the Reference for information on the pgx_dmpall command and pgx_rcvall command.

Setting a restore point

In case you want to recover your database to a certain point in time, you can name this particular point in time, which is referred to as the restore point, by using the psql command.

By setting a restore point before executing an application, it becomes easy to identify up to which point in time the data will be reverted.

A restore point can be set to any point in time after a backup is executed. However, if a restore point is set before a backup is executed, the database cannot be recovered to that point in time. This is because restore points are recorded in the archive logs, and the archive logs are discarded when backups are executed.

Example

The following example uses the psql command to connect to the database and execute the SQL statement to set a restore point.

However, when considering continued compatibility of applications, do not use functions directly in SQL statements. Refer to "Notes on Application Compatibility" in the Application Development Guide for details.

```
postgres=# SELECT pg_create_restore_point('batch_20220303_1');
LOG:  restore point "batch_20220303_1" created at 0/20000E8
STATEMENT:  select pg_create_restore_point('batch_20220303_1');
 pg_create_restore_point
-------------------------
 0/20000E8
(1 row)
```

Refer to "16.3.2 Using the pgx_rcvall Command" for information on using a restore point to recover the database.

Note

- Name restore points so that they are unique within the database. Add the date and time of setting a restore point to distinguish it from other restore points, as shown below:
  - YYMMDD_HHMMSS
    - YYMMDD: Indicates the date
    - HHMMSS: Indicates the time
- There is no way to check restore points you have set. Keep a record in, for example, a file.

See

Refer to "System Administration Functions" under "Functions and Operators" in the PostgreSQL Documentation for information on pg_create_restore_point.

Chapter 4 Configuring Secure Communication Using Secure Sockets Layer

If communication data transferred between a client and a server contains confidential information, encrypting the communication data can protect it against threats, such as eavesdropping on the network.

4.1 Configuring Communication Data Encryption

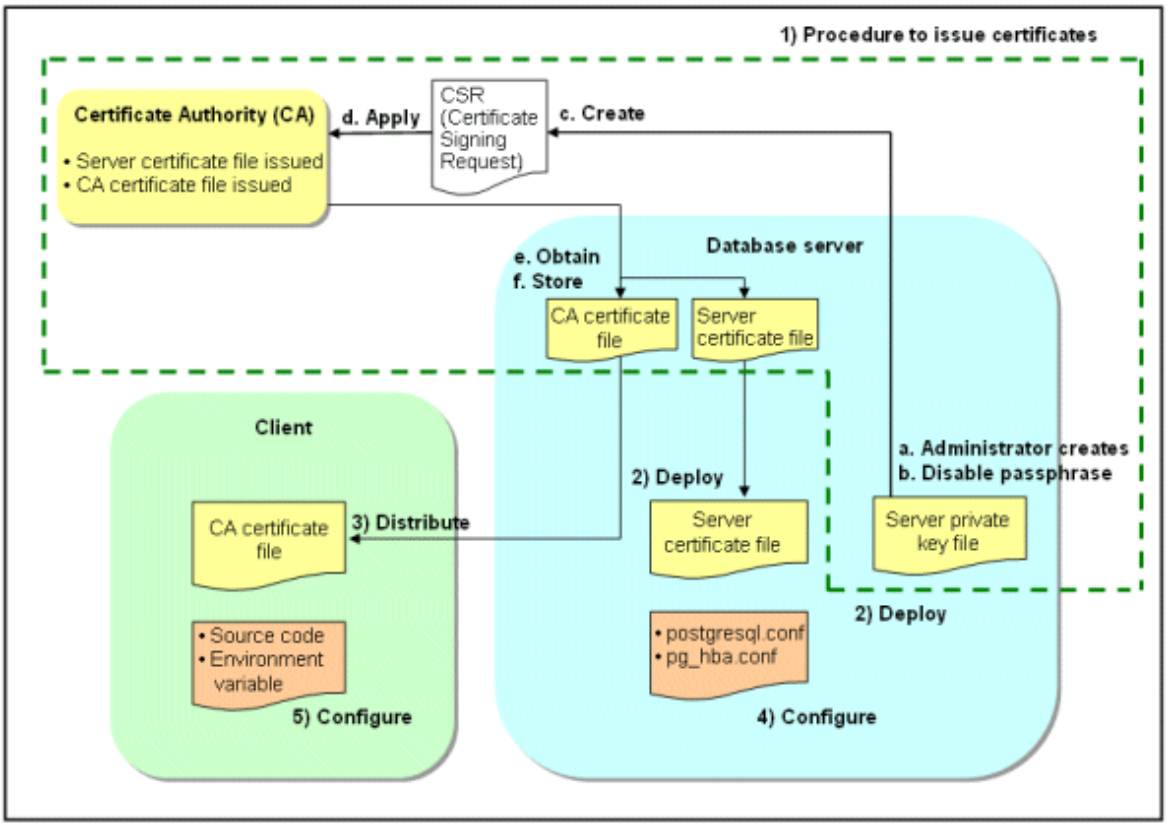
To encrypt communication data transferred between a client and a server, configure communication data encryption as described below. Communication data encryption not only protects the communication content, but it also guards against man-in-the-middle (MITM) attacks (for example, data and password theft through server impersonation).

Table 4.1 Configuration procedure

Configuration procedure
1) Issue a certificate
2) Deploy a server certificate file and a server private key file
3) Distribute a CA certificate file to the client
4) Configure the operating environment for the database server
5) Configure the operating environment for the client

The following figure illustrates the environment for communication data encryption.

Figure 4.1 Environment for communication data encryption



4.1.1 Issuing a Certificate

For authenticating servers, you must acquire a certificate issued by the certificate authority (CA).
Fujitsu Enterprise Postgres supports X.509 standard PEM format files. If the certificate authority issues a file in DER format, use a tool such as the openssl command to convert the DER format file to PEM format.

The following provides an overview of the procedure. Refer to the procedure published by the public or independent certificate authority (CA) that provides the certificate file for details.

a. Create a server private key file

b. Disable the passphrase for the server private key file

c. Create a CSR (signing request for obtaining a server certificate) from the server private key file

d. Apply to the certificate authority (CA) for a server certificate

e. Obtain a server certificate file and a CA certificate file from the certificate authority (CA)

f. Store the server certificate file and the CA certificate file
   Note: If you lose or destroy the certificates, you will need to have them re-issued.

The above procedure enables you to prepare the following files:

- Server private key file
- Server certificate file
- CA certificate file

4.1.2 Deploying a Server Certificate File and a Server Private Key File

Create a directory on the local disk of the database server and store the server certificate file and the server private key file in it.
Use the operating system features to set access privileges for the server certificate file and the server private key file so that only the database administrator has load privileges.
Back up the server certificate file and the server private key file in the event that data corruption occurs and store them securely.

4.1.3 Distributing a CA Certificate File to the Client

Create a directory on the local disk of the client and place the distributed CA certificate file there. Use the operating system features to set load privileges to protect the CA certificate file against accidental deletion.

4.1.4 Configuring the Operating Environment for the Database Server

See

Refer to "Secure TCP/IP Connections with SSL" under "Server Administration" in the PostgreSQL Documentation for details.

4.1.5 Configuring the Operating Environment for the Client

See

Refer to the following sections in the Application Development Guide for details, depending on your application development environment:

- "Settings for Encrypting Communication Data" under "Setup" in "JDBC Driver"
- "Settings for Encrypting Communication Data" under "Setup" in "C Library (libpq)"
- "Settings for Encrypting Communication Data" under "Setup" in "Embedded SQL in C"

4.1.6 Performing Database Multiplexing

When you perform communication that uses database multiplexing and a Secure Socket Layer server certificate, certificates with the same "Common Name" must be used. To ensure this, take one of the following actions:

- Create one server certificate, replicate it, and place a copy on each server used for database multiplexing.
- Create a server certificate with the same "Common Name" for each server used for database multiplexing.

See

Refer to "Using the Application Connection Switch Feature" in the Application Development Guide for information on how to specify applications on the client.

Chapter 5 Protecting Storage Data Using Transparent Data Encryption

This chapter describes how to encrypt data to be stored in the database.

See

If you want to use an external key management system as the storage location for the encryption key, refer to "Chapter 6 Using Transparent Data Encryption with Key Management Systems as Keystores".

5.1 Protecting Data Using Encryption

With PostgreSQL, data in a database is protected from access by unauthorized database users through the use of authentication and access controls. However, the OS file is not protected from attackers who bypass the database server's authentication and access controls.

With Fujitsu Enterprise Postgres, data inside the OS file is encrypted, so valuable information is protected even if the file or disk is stolen.

Data to be stored in a database is encrypted when it is written to the data file, and decrypted when it is read.

This is performed automatically by the instance, so the user and the application need not be aware of key management and encryption or decryption. This process is called TDE (Transparent Data Encryption).

The characteristics of TDE are described below.

Encryption mechanisms

Two-layer encryption key and the keystore

In each tablespace, there is a tablespace encryption key that encrypts and decrypts all the data within. The tablespace encryption key is encrypted by the master encryption key and saved.
Only one master encryption key exists in a database cluster. It is encrypted based on a passphrase specified by the user and stored in a keystore. Fujitsu Enterprise Postgres provides a file-based keystore. Attackers who do not know the passphrase cannot read the master encryption key from the keystore.

Strong encryption algorithms

TDE uses the Advanced Encryption Standard (AES) as its encryption algorithm. AES was adopted as a standard in 2002 by the United States Federal Government, and is used throughout the world.

Faster encryption and decryption based on hardware

TDE minimizes the overhead of encryption and decryption by using the AES-NI (Advanced Encryption Standard New Instructions) built into Intel(R) Xeon(R) processors since the 5600 series. This means that even in situations where previously the minimum encryption target was selected as a tradeoff between performance and security, it is now possible to encrypt all the data of an application.

Refer to the Intel Corporation's website for information on the list of processors equipped with AES-NI.

Zero overhead storage areas

Encryption does not change the size of data stored in tables, indexes, or WAL. There is, therefore, no need for additional estimates or disks.

Scope of encryption

All user data within the specified tablespace

The tablespace is the unit for specifying encryption. All tables, indexes, temporary tables, and temporary indexes created in the encrypted tablespace are encrypted. There is no need for the user to consider which tables and strings to encrypt.

Refer to "5.4 Encrypting a Tablespace" for details.

Backup data

The pgx_dmpall command and pg_basebackup command create backup data by copying the OS file. Backups of the encrypted data are, therefore, also encrypted. Information is protected from leakage even if the backup medium is stolen.

WAL and temporary files

WAL, which is created by updating encrypted tables and indexes, is encrypted with the same security strength as the update target. When large merges and sorts are performed, the encrypted data is written to a temporary file in encrypted format.

Streaming replication support

You can combine streaming replication and transparent data encryption. The data and WAL encrypted on the primary server is transferred to the standby server in its encrypted format and stored.

Note

The following are not encrypted:

- pg_dump and pg_dumpall output files
- Files output by the COPY command
- Notification event payloads that communicate using the LISTEN or NOTIFY command
- Checksum validation is not performed on encrypted tablespaces during backup and when using the pg_checksum utility.

5.2 Setting the Master Encryption Key

To use transparent data encryption, you must create a keystore and set the master encryption key.

1. In the keystore_location parameter of postgresql.conf, specify the directory to store the keystore.

   Specify a different location for each database cluster.

   ```
   keystore_location = '/key/store/location'
   ```

   Refer to "Appendix A Parameters" for information on postgresql.conf.

   After editing the postgresql.conf file, either start or restart the instance.

   - Using WebAdmin

     Refer to "2.1.1 Using WebAdmin", and restart the instance.

   - Using the pg_ctl command

     Specify the following in the pg_ctl command:

     - Specify "restart" as the mode.
     - Specify the data storage destination directory in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
     - Specify the -w option. This means that the command returns after waiting for the instance to start. If the -w option is not specified, it may not be possible to determine if the starting of the instance completed successfully or if it failed.

     Example

     ```
     > pg_ctl restart -w -D /database/inst1
     ```

2. Execute an SQL function, such as the one below, to set the master encryption key. This must be performed by the superuser. Execute it as the database superuser.

   ```
   SELECT pgx_set_master_key('passphrase');
   ```

   The value "passphrase" is the passphrase that will be used to open the keystore. The master encryption key is protected by this passphrase, so avoid specifying a short simple string that is easy to guess.

   Refer to "B.2 Transparent Data Encryption Control Functions" for information on the pgx_set_master_key function.

Note

Note that if you forget the passphrase, you will not be able to access the encrypted data. There is no method to retrieve a forgotten passphrase and decrypt data. Do not, under any circumstances, forget the passphrase.

The pgx_set_master_key function creates a file with the name keystore.ks in the keystore storage destination. It also creates a master encryption key from random bit strings, encrypts it with the specified passphrase, and stores it in keystore.ks. At this point, the keystore is open.

5.3 Opening the Keystore

To create encrypted tablespaces and access the encrypted data, you must first open the keystore. When you open the keystore, the master encryption key is loaded into the database server memory and becomes usable for encryption and decryption.

You need to open the keystore each time you start the instance. To open the keystore, the database superuser must execute the following SQL function.

```
SELECT pgx_open_keystore('passphrase');
```

The value "passphrase" is the passphrase specified during creation of the keystore.

Refer to "B.2 Transparent Data Encryption Control Functions" for information on the pgx_open_keystore function.

Note that, in the following cases, the passphrase must be entered when starting the instance, because the encrypted WAL must be decrypted for recovery. In this case, the above-mentioned pgx_open_keystore function cannot be executed.

- If performing crash recovery at the time of starting the instance
- If performing recovery using continuous archiving

For the above cases, specify the --keystore-passphrase option in the pg_ctl command, and then start the instance. This will display the prompt for the passphrase to be entered, as shown below.

```
> pg_ctl --keystore-passphrase start
Enter the passphrase:
The server is starting
>
```

Point

When using an automatically opening keystore, you do not need to enter the passphrase and you can automatically open the keystore when the database server starts. Refer to "5.6.3 Enabling Automatic Opening of the Keystore" for details.

5.4 Encrypting a Tablespace

The keystore must be open before you can create an encrypted tablespace.

When creating a tablespace that will be encrypted, configure the encryption algorithm in the runtime parameters. For example, to create a tablespace with the name secure_tablespace using AES with a key length of 256 bits as the encryption algorithm, configure as shown below.

```
-- Specify the encryption algorithm for the tablespace to be created below
SET tablespace_encryption_algorithm = 'AES256';
CREATE TABLESPACE secure_tablespace LOCATION '/My/Data/Dir';
-- Specify that the tablespace to be created below is not to be encrypted
SET tablespace_encryption_algorithm = 'none';
```

Or

```
CREATE TABLESPACE secure_tablespace LOCATION '/My/Data/Dir' WITH (tablespace_encryption_algorithm =
'AES256' );
```

When the tablespace is empty, the encryption algorithm can be modified with the command below.

```
ALTER TABLESPACE secure_tablespace SET (tablespace_encryption_algorithm=AES256);
```

Trying to set the encryption algorithm for a non-empty tablespace causes an error.

You can use AES with a key length of 128 bits or 256 bits as the encryption algorithm. It is recommended that you use 256-bit AES. Refer to "Appendix A Parameters" for information on how to specify the runtime parameters.

If user provides both GUC and command line options while creating the tablespace, the preference is given to the command line option.

The pg_default and pg_global tablespaces cannot be encrypted.

Create tables and indexes in the encrypted tablespace that you created. Relations created in the encrypted tablespace are automatically encrypted.

Example

Example 1: Specifying an encrypted tablespace when creating it

```
CREATE TABLE my_table (...)
    TABLESPACE secure_tablespace;
```

Example 2: Not explicitly specifying a tablespace when creating it and instead using the default tablespace

```
SET default_tablespace = 'secure_tablespace';
CREATE TABLE my_table (...);
```

The process is the same for encrypting temporary tables and temporary indexes. In other words, either explicitly specify the TABLESPACE clause or list encrypted tablespaces in the temp_tablespaces parameter, and then execute CREATE TEMPORARY TABLE or CREATE INDEX.

Point

If an encrypted tablespace is specified in the TABLESPACE clause of the CREATE DATABASE statement, relations created in the database without explicitly specifying a tablespace will be encrypted. Furthermore, the system catalog will also be encrypted, so the source code of user-defined functions is also protected.

Example: Specifying a tablespace in a database definition statement

```
CREATE DATABASE DB01 TABLESPACE=SP01 ... ;
```

Part of the data is also stored in the system catalog - to encrypt this data as well, specify an encrypted tablespace as above and create a database.

Note

An encrypted tablespace cannot be created from the window used for creating the pgAdmin tablespace. To create an encrypted tablespace, create an encrypted tablespace in the Query tool.

5.5 Checking an Encrypted Tablespace

The pgx_tablespaces system view displays information about whether each tablespace has been encrypted, and about the encryption algorithm. Refer to "C.1 pgx_tablespaces" for information on strings.

You can discover which tablespaces have been encrypted by executing the following SQL statements.

However, when considering continued compatibility of applications, do not reference system catalogs (pg_tablespace) directly in SQL statements.

```
SELECT spcname, spcencalgo
FROM pg_tablespace ts, pgx_tablespaces tsx
WHERE ts.oid = tsx.spctablespace;
```

Example

```
postgres=# SELECT spcname, spcencalgo FROM pg_tablespace ts, pgx_tablespaces tsx WHERE ts.oid =
tsx.spctablespace;
      spcname      | spcencalgo
-------------------+------------
 pg_default        | none
 pg_global         | none
 secure_tablespace | AES256
(3 rows)
```

See

Refer to "Notes on Application Compatibility" in the Application Development Guide for information on how to maintain application compatibility.

5.6 Managing the Keystore

This section describes how to manage the keystore and the master encryption key to guard against the threat of theft.

5.6.1 Changing the Master Encryption Key

Using the same encryption key for an extended period gives attackers an opportunity to decipher the encrypted data. It is recommended that you change the key at regular intervals, or whenever the key is exposed to risk.

Adhere to the industry's best practices for encryption algorithms and key management when considering how often the key should be changed. For example, the NIST in the United States has published "NIST Special Publication 800-57". The PCI DSS also refers to this publication. This publication recommends changing the master encryption key once a year.

To change the master encryption key, execute the pgx_set_master_key function, which is the same function used for configuring the key. Refer to "5.2 Setting the Master Encryption Key" for details.

After changing the master encryption key, you must immediately back up the keystore.

5.6.2 Changing the Keystore Passphrase

In security policies for organizations, it is usually a requirement that the passphrase be changed whenever a security administrator who knows the passphrase is removed from duties due to transfer or retirement. It is also recommended that the passphrase be changed if it is ever exposed to risks due to deception such as social engineering.

To change the keystore passphrase, execute the following SQL function as a superuser.

```
SELECT pgx_set_keystore_passphrase('oldPassphrase', 'newPassphrase');
```

After changing the passphrase, you must immediately back up the keystore.

Refer to "B.2 Transparent Data Encryption Control Functions" for information on the pgx_set_keystore_passphrase function.

5.6.3 Enabling Automatic Opening of the Keystore

When using an automatically opening keystore, you do not need to enter the passphrase and you can automatically open the keystore when the instance starts. Execute the pgx_keystore command to enable automatic opening of the keystore.

```
> pgx_keystore --enable-auto-open /key/store/location/keystore.ks
Enter the passphrase:
Automatic opening of the keystore is now enabled
>
```

See

Refer to "pgx_keystore" in the Reference for information on pgx_keystore command.

When automatic opening is enabled, an automatically opening keystore is created in the same directory as the original keystore. The file name of the automatically opening keystore is keystore.aks. The file keystore.aks is an obfuscated copy of the decrypted content of the keystore.ks file. As long as this file exists, there is no need to enter the passphrase to open the keystore when starting the instance.

Do not delete the original keystore file, keystore.ks. It is required for changing the master encryption key and the passphrase. When you change the master encryption key and the passphrase, keystore.aks is recreated from the original keystore file, keystore.ks.

Protect keystore.ks, keystore.aks, and the directory that stores the keystore so that only the user who starts the instance can access them.

Configure the permission of the files so that only the user who starts the instance can access the SQL functions and commands that create these files. Accordingly, manually configure the same permission mode if the files are restored.

Example

```
# chown -R fsepuser:fsepuser /key/store/location
# chmod 700 /key/store/location
# chmod 600 /key/store/location/keystore.ks
# chmod 600 /key/store/location/keystore.aks
```

An automatically opening keystore will only open on the computer where it was created.

To disable automatic opening of the keystore, delete keystore.aks.

Note

- To use WebAdmin for recovery, you must enable automatic opening of the keystore.
- Refer to "5.7 Backing Up and Restoring/Recovering the Database" after enabling or reconfiguring encryption to back up the database.
- Specify a different directory from those below as the keystore storage destination:
  - Data storage destination
  - Tablespace storage destination
  - Transaction log storage destination
  - Backup data storage destination

5.6.4 Backing Up and Recovering the Keystore

Back up the keystore at the following times in case it is corrupted or lost. Note that you must store the database and the keystore on separate data storage media. Storing both on the same data storage medium risks the danger of the encrypted data being deciphered if the medium is stolen. A passphrase is not required to open an automatically opening keystore, so store this type of keystore in a safe location.

- When the master encryption key is first configured
- When the master encryption key is changed
- When the database is backed up
- When the keystore passphrase is changed

Point

Do not overwrite an old keystore when backing up a keystore. This is because during database recovery, you must restore the keystore to its state at the time of database backup. When the backup data of the database is no longer required, delete the corresponding keystore.

Example

- Back up the database and the keystore on March 1, 2022.

```
> pgx_dmpall -D /database/inst1
> cp -p /key/store/location/keystore.ks /keybackup/keystore_20220301.ks
```

  Specify the following in the pgx_dmpall command:

  - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

- Change the master encryption key, and back up the keystore on March 5, 2022.

```
> psql -c "SELECT pgx_set_master_key('passphrase')" postgres
> cp -p /key/store/location/keystore.ks /keybackup/keystore_20220305.ks
```

  Specify the following in the psql command:

  - Specify the SQL function that sets the master encryption key in the -c option.
  - Specify the name of the database to be connected to as the argument.

If the keystore is corrupted or lost, restore the keystore containing the latest master encryption key. If there is no keystore containing the latest master encryption key, restore the keystore to its state at the time of database backup, and recover the database from the database backup. This action recovers the keystore to its latest state.

Example

- Restore the keystore containing the latest master encryption key as of March 5, 2022.

```
> cp -p /keybackup/keystore_20220305.ks /key/store/location/keystore.ks
```

- If there is no backup of the keystore containing the latest master encryption key, recover the keystore by restoring the keystore that was backed up along with the database on 1 March 2022.

```
> cp -p /keybackup/keystore_20220301.ks /key/store/location/keystore.ks
> pgx_rcvall -B /backup/inst1 -D /database/inst1 --keystore-passphrase
```

  Specify the following in the pgx_rcvall command:

  - Specify the data storage directory in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
  - Specify the backup data storage directory in the -B option.
  - The --keystore-passphrase option prompts you to enter the passphrase to open the keystore.

If you have restored the keystore, repeat the process of enabling automatic opening of the keystore. This ensures that the contents of the automatically opening keystore (keystore.aks) are identical to the contents of the restored keystore.

It is recommended that you do not back up the automatically opening keystore file, keystore.aks. If the database backup medium and the backup medium storing the automatically opening keystore are both stolen, the attacker will be able to read the data even without knowing the passphrase.

If the automatically opening keystore is corrupted or lost, you must again enable automatic opening. The keystore.aks file will be recreated from keystore.ks at this time.

See

Refer to "pgx_rcvall" and "pgx_dmpall" in the Reference for information on the pgx_rcvall and pgx_dmpall commands.

Refer to "psql" under "Reference" in the PostgreSQL Documentation for information on the psql command.

Refer to "B.2 Transparent Data Encryption Control Functions" for information on the pgx_set_master_key function.

Refer to "5.6.3 Enabling Automatic Opening of the Keystore" for information on how to enable automatic opening of the keystore.

5.7 Backing Up and Restoring/Recovering the Database

Fujitsu Enterprise Postgres enables you to use the five backup and recovery methods described below. Regardless of the method you use, you must back up the keystore at the same time.

Note that you must store the database and the keystore on separate data storage media. Storing both on the same data storage medium risks the danger of the encrypted data being deciphered if the medium is stolen.

Backup and recovery using WebAdmin

- Backup

  WebAdmin backs up encrypted data.

  Back up the key store after backing up the database.

- Recovery

  Restore the keystore to its state at the time of database backup. Refer to "5.6.4 Backing Up and Recovering the Keystore" for details.

  Enable automatic opening of the keystore in accordance with the procedure described in "5.6.3 Enabling Automatic Opening of the Keystore". Then, use WebAdmin to recover the database.

Backup and recovery using the pgx_dmpall and pgx_rcvall commands

- Backup

  The pgx_dmpall command backs up the encrypted data.

  Back up the key store after backing up the database.

- Recovery

  Restore the keystore to its state at the time of the database backup.

  Configure automatic opening of the key store as necessary.

  If automatic opening of the keystore is not enabled, execute the pgx_rcvall command with the --keystore-passphrase option specified. This will display the prompt for the passphrase to be entered.

Example

- Back up the database and the keystore on March 1, 2022.

```
> pgx_dmpall -D /database/inst1
> cp -p /key/store/location/keystore.ks /keybackup/keystore_20220301.ks
```

  Specify the following in the pgx_dmpall command:

  - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

- Recover the database and the keystore from the backup taken on March 1, 2022.

```
> cp -p /keybackup/keystore_20220301.ks /key/store/location/keystore.ks
> pgx_keystore --enable-auto-open /key/store/location/keystore.ks  (Execute only when enabling
automatic opening)
> pgx_rcvall -B /backup/inst1 -D /database/inst1 --keystore-passphrase
```

  Specify the following in the pgx_rcvall command:

  - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
  - Specify the backup data storage directory in the -B option.
  - The --keystore-passphrase option prompts you to enter the passphrase to open the keystore.

Dump and restore using SQL

- Backup

  The files output by the pg_dump and pg_dumpall commands are not encrypted. You should, therefore, encrypt the files using OpenSSL commands or other means before saving them, as described in "5.8 Importing and Exporting the Database" below.

  Back up the key store after backing up the database.

- Restore

  If the backup data has been encrypted using, for example Open SSL commands, decrypt that data.

  The data generated by the pg_dumpall command includes a specification to encrypt tablespaces by default. For this reason, the psql command encrypts tablespaces during restoration.

File system level backup and restore

- Backup

  Stop the instance and backup the data directory and the tablespace directory using the file copy command of the operating system. The files of encrypted tablespaces are backed up in the encrypted state.

  Back up the key store after performing the backup.

- Restore

  Restore the keystore to its state at the time of the database backup.

  Stop the instance and restore the data directory and the tablespace directory using the file copy command of the operating system.

Continuous archiving and point-in-time recovery

- Backup

  The pg_basebackup command backs up the encrypted data as is.

  Back up the key store after performing the backup.

- Recovery

  Restore the keystore to its state at the time of the database backup.

  Configure automatic opening of the key store as necessary.

  If automatic opening of the keystore is not enabled, execute the pg_ctl command to start the instance with the --keystore-passphrase option specified. This will display the prompt for the passphrase to be entered.

See

- Refer to "pg_ctl" under "Reference" in the PostgreSQL Documentation for information on the pg_ctl command.

- Refer to "Reference" in the PostgreSQL Documentation for information on the following commands:
  - psql
  - pg_dump
  - pg_basebackup
- Refer to the Reference for information on the following commands:
  - pgx_rcvall
  - pgx_dmpall
  - pg_dumpall

If you have restored the keystore, repeat the process of enabling automatic opening of the keystore This ensures that the contents of the automatically opening keystore (keystore.aks) are identical to the contents of the restored keystore.

Refer to "5.6.3 Enabling Automatic Opening of the Keystore" for information on how to enable automatic opening of the keystore.

5.8 Importing and Exporting the Database

The files output by the COPY TO command are not encrypted. Therefore, when transferring files to other systems, you should encrypt files using OpenSSL commands or other means and use scp or sftp to encrypt the data being transferred.

Use a safe method to delete obsolete plain text files.

You can use the following methods to safely delete files:

- shred command

Example

```
# Export the contents of the table my_table to a CSV file.
> psql -c "COPY my_table TO '/tmp/my_table.csv' (FORMAT CSV)" postgres

# Encrypt the exported file.
> openssl enc -e -aes256 -in my_table.csv -out my_table.csv.enc
(The user is prompted to enter the passphrase to be used for encryption)

# Safely delete plain text files.
> shred -u -x my_table.csv
 (Transfer encrypted files to other systems)

# Decrypt the encrypted files on other systems.
> openssl enc -d -aes256 -in my_table.csv.enc -out my_table.csv
(The user is prompted to enter the passphrase to be used for decryption)
```

If you use COPY FROM to import data to tables and indexes in an encrypted tablespace, the imported data is automatically encrypted before being stored.

5.9 Encrypting Existing Data

You cannot encrypt existing unencrypted tablespaces. In addition, you cannot change encrypted tablespaces so that they do not encrypt.

As an alternative, transfer the tables and indexes to other tablespaces. You can use the following SQL commands for this.

```
ALTER TABLE table_name SET TABLESPACE new_tablespace;
ALTER INDEX index_name SET TABLESPACE new_tablespace;
ALTER DATABASE database_name SET TABLESPACE new_tablespace;
```

See

Refer to "SQL Commands" under "Reference" in the PostgreSQL Documentation for information on SQL commands.

5.10 Operations in Cluster Systems

This section describes how to use transparent data encryption on cluster systems such as high-availability systems, streaming replication, and database multiplexing.

5.10.1 HA Clusters that do not Use Database Multiplexing

Take the following points into account when using transparent data encryption in an HA cluster environment that does not use database multiplexing.

Placement and automatic opening of the keystore file

There are two alternatives for placing the keystore file:

- Sharing the keystore file
- Placing a copy of the keystore file

Sharing the keystore file

This involves using the same keystore file on the primary server and the standby server.

As the standby server is not active while the primary server is running, this file would not be accessed simultaneously, and therefore, it can be shared.

To manage the keystore file in a more secure manner, place it on the key management server or the key management storage isolated in a secure location.

Enable the automatic opening of the keystore on both the primary and standby servers.

Placing a copy of the keystore file

This involves placing a copy of the primary server keystore file on the standby server.

You can do this if you cannot prepare a shared server or disk device that can be accessed from both the primary and standby servers.

However, if you change the master encryption key and the passphrase on the primary server, you must copy the keystore file to the standby server again.

To manage the keystore file in a more secure manner, prepare the key management server or the key management storage isolated in a secure location for both the primary and standby servers, and place the keystore files there.

Enable the automatic opening of the keystore on both the primary and standby servers. Note that copying the automatically opening keystore file (keystore.aks) to the standby server does not enable the automatic opening of the keystore.

See

Refer to the Cluster Operation Guide (PRIMECLUSTER) for information on building a cluster system environment for performing failover using the failover feature integrated with the cluster software.

5.10.2 Database Multiplexing Mode

Note the following when using transparent data encryption in environments that use streaming replication, or database multiplexing with streaming replication.

Placing the keystore file

Place a copy of the primary server keystore file on the standby server.

This is required as the keystore file cannot be shared, and both servers may need to access it simultaneously.

Point

To manage the keystore file in a more secure manner, place it on the key management server or the key management storage isolated in a secure location. A keystore used by both the primary and standby servers can be managed on the same key management server or key management storage.

However, create different directories for the keystores to be used by the primary server and the standby server. Then copy the keystore for the primary server to the directory used on the standby server.

Automatically opening the keystore

You must enable automatic opening of the keystore.

To do this, enable automatic opening of the keystore in all servers that make up database multiplexing. The settings for automatic opening of the keystore include information unique to each server, so simply copying the file does not enable it.

Changing the passphrase

Changes to the passphrase are reflected in all servers that make up database multiplexing, so no special operation is required.

Building and starting a standby server

Before using the pg_basebackup command or pgx_rcvall command to build a standby server, copy the keystore file from the primary server to the standby server. When using an automatically opening keystore, use the copied keystore file to enable automatic opening on the standby server.

Open the keystore each time you start the standby server. This step is necessary for decrypting and restoring encrypted WAL received from the primary server. To open the keystore, specify the --keystore-passphrase option in the pg_ctl command or pgx_rcvall command and enter the passphrase, or use an automatically opening keystore.

Changing the master encryption key and the passphrase

Change the master encryption key and the passphrase on the primary server. You need not copy the keystore from the primary server to the standby server. You need not even restart the standby server or reopen the keystore. Changes to the master encryption key and the passphrase are reflected in the keystore on the standby server.

See

Refer to "pgx_rcvall " in the Reference for information on pgx_rcvall command.

Refer to "pg_ctl" under "Reference" in the PostgreSQL Documentation for information on pg_ctl command.

Refer to "pg_basebackup" under "Reference" in the PostgreSQL Documentation for information on pg_basebackup command.

Refer to "High Availability, Load Balancing, and Replication" under "Server Administration" in the PostgreSQL Documentation for information on how to set up streaming replication.

5.11 Security-Related Notes

- Decrypted data is cached in the database server memory (shared buffer). As a result, unencrypted data is stored in a core file, which is a process memory dump. You should, therefore, safely delete the memory dump.
  You can safely delete files by using the following command:
  - shred command
- Unencrypted data may be written from the database server memory to the operating system's swap area. To prevent leakage of information from the swap area, consider either disabling the use of swap area or encrypting the swap area using a full-disk encryption product.

- The content of the server log file is not encrypted. Therefore, in some cases the value of a constant specified in a SQL statement is output to the server log file. To prevent this, consider setting a parameter such as log_min_error_statement.
- When executing an SQL function that opens the keystore and modifies the master encryption key, ensure that the SQL statement containing the passphrase is not output to the server log file. To prevent this, consider setting a parameter such as log_min_error_statement. If you are executing this type of SQL function on a different computer from the database server, encrypt the communication between the client and the database server with SSL.
- Starting with FEP 10, logical replication is available, which allows non-backed up clusters to subscribe to databases where transparent data encryption is enabled. Logical replication does not need to have the same encryption strategy between publisher and subscriber.

  In this scenario, if the user wants to encrypt the subscribed copy of data as well, then it is the user's responsibility to create encryption policies to the subscribed databases. By default, published encrypted tablespace data will not be encrypted in the subscriber side.

5.12 Tips for Installing Built Applications

With transparent data encryption, you can easily encrypt all the data in an application without modifying the application. Database administrators install built applications in the following manner. However, this procedure stores data to the default tablespace, so take necessary action if processing differs from the original design.

1. (Normal procedure) Create an owner and a database for the built application.

```
CREATE USER crm_admin ...;
CREATE DATABASE crm_db ...;
```

2. (Procedure for encryption) Create an encrypted tablespace to store the data for the built application.

```
SET tablespace_encryption_algorithm = 'AES256';
CREATE TABLESPACE crm_tablespace LOCATION '/crm/data';
```

3. (Procedure for encryption) Configure an encrypted tablespace as the default tablespace for the owner of the built application.

```
ALTER USER crm_admin SET default_tablespace = 'crm_tablespace';
ALTER USER crm_admin SET temp_tablespaces = 'crm_tablespace';
```

4. (Normal procedure) Install the built application. The application installer prompts you to enter the host name and the port number of the database server, the user name, and the database name. The installer uses the entered information to connect to the database server and execute the SQL script. For applications that do not have an installer, the database administrator must manually execute the SQL script.

Normally, the application's SQL script includes logic definition SQL statements, such as CREATE TABLE, CREATE INDEX, and GRANT or REVOKE, converted from the entity-relationship diagram. It does not include SQL statements that create databases, users, and tablespaces. Configuring the default tablespace of the users who will execute the SQL script deploys the objects generated by the SQL script to the tablespace.

Chapter 6 Using Transparent Data Encryption with Key Management Systems as Keystores

This chapter describes the operation of transparent data encryption when a key management system is used as a keystore.

Note

This feature is not available on SLES 12.

See

Refer to "Key Management System Requirements" in the Installation and Setup Guide for Server for the key management system requirements that can be used with Fujitsu Enterprise Postgres.

6.1 Protecting Data Using Encryption

Refer to "5.1 Protecting Data Using Encryption". The following describes the differences from the transparent data encryption operation in the file-based keystore described in "5.1 Protecting Data Using Encryption".

Encryption mechanisms

Two-layer encryption key and the keystore

Each tablespace has a tablespace encryption key that encrypts/decrypts all data in it. Tablespace encryption keys are stored encrypted with the master encryption key.
Use an encryption key stored in a key management system as a common master encryption key for your database cluster. Fujitsu Enterprise Postgres refers to the key management system as a keystore for master encryption keys.

Type of key management system

Two types of key management systems are available:

- kmip

  It is a key management system that can be used using a protocol called KMIP (Key Management Interoperability Protocol) standardized by OASIS (Organization for the Advancement of Structured Information Standards).

- custom

  It is a key management system that cooperates using an adapter that converts the request format without adopting the KMIP protocol.

See

Refer to "Key Management System Requirements" in the Installation and Setup Guide for Server for the key management system requirements that can be used with Fujitsu Enterprise Postgres.

Sharing tablespace encryption keys

When using an adapter to link with a key management system, encryption and decryption of the tablespace encryption key using the master encryption key are performed on the key management system side.

Tablespace encryption keys can be shared within a database cluster so that you do not need to access the key management system each time you want to use the tablespace encryption key.

The cost of encryption/decryption using the master encryption key becomes an issue in the following cases:

- Multiple connections to the database access encrypted tablespaces

- Connections accessing encrypted tablespaces are repeated and connection pooling is disabled

Encryption key identifier

Key IDs are used as identifiers to identify encryption keys stored on the key management system.

Key ID

Information that identifies the encryption key on the key management system, and is unique within the key management system. The correspondence between the encryption key substance (byte string) and the key ID does not change throughout the life cycle of the encryption key.

The name of the identifier differs depending on each key management system, but in this feature, such information is called the key ID.

Changes to the key management system

After starting operation of the transparent data encryption function, the key management system to be used can be changed to another key management system.

6.2 Setting the Master Encryption Key

To use transparent data encryption, you must create a keystore and set the master encryption key.

1. Load the shared_preload_libraries parameter in postgresql.conf with the library name "tde _ kms"

```
shared_preload_libraries = 'tde_kms'
```

2. When using an adapter, register the adapter as a plug-in. Specify the directory where the plugin is stored in the tde_kms.plugin_path parameter in postgresql.conf. Store your plugins in this directory.

```
tde_kms.plugin_path = '/home/fsepuser/plugin/'
```

3. To share the tablespace encryption key, set the tde_kms.enable_shared_dek parameter in postgresql.conf to "on".

```
tde_kms.enable_shared_dek = on
```

4. Set the tde_kms.kms_conninfo_file parameter in postgresql.conf to a file that contains key management system connection information. Refer to "Appendix A Parameters" for information.

   Example for the key management system connection information file kms_conninfo.conf

```
tde_kms.kms_conninfo_file = 'kms_conninfo.conf'
```

   Example of key management system connection information file

   For type kmip

```
kmip   mykmipsvr  mykmipsvr.example.com   5696   cert   sslcert=postgres.crt
sslkey=postgres.key sslrootcert=root.crt
```

   For type custom

```
custom   mykms   mykms  arg=--profile arg=user1
```

5. Execute a CREATE EXTENSION statement to install the extension.

```
CREATE EXTENSION tde_kms;
```

6. To enable transparent data encryption, call the pgx_declare_external_master_key function to declare the encryption key to use as the master encryption key. Specify a key ID as an identifier to identify the encryption key. Refer to "B.2.3 pgx_declare_external_master_key" for information on the pgx_declare_external_master_key function.

```
SELECT pgx_declare_external_master_key( kms_name => 'mykmipsvr', key_id =>
'a0eebc99-9c0b-0000-0000-000000000000', sslpassphrase => 'mykmippassphrase' );
```

6.3 Opening the Keystore

To create encrypted tablespaces and access the encrypted data, you must first open the keystore. When you open the keystore, the master encryption key is available for encryption and decryption.

You need to open the keystore each time you start the instance. To open the keystore, the database superuser must execute the following SQL function.

```
SELECT pgx_open_keystore( sslpassphrase => 'passphrase');
```

passphrase is the passphrase of the private key file for the client certificate.

Refer to "B.2 Transparent Data Encryption Control Functions" for information on the pgx_open_keystore function.

Note that, in the following cases, the passphrase must be entered when starting the instance, because the encrypted WAL must be decrypted for recovery. In this case, the above-mentioned pgx_open_keystore function cannot be executed.

- If performing crash recovery at the time of starting the instance
- If performing recovery using continuous archiving

For the above cases, specify the --kms-secret option in the pg_ctl command, and then start the instance. This will display the prompt for the passphrase to be entered, as shown below.

```
> pg_ctl --kms-secret start
Enter secret:
```

Point

When using an automatically opening keystore, you do not need to enter the passphrase and you can automatically open the keystore when the database server starts. Refer to "6.6.2 Enabling Automatic Opening of the Keystore" for details.

6.4 Encrypting a Tablespace

Refer to "5.4 Encrypting a Tablespace".

6.5 Checking an Encrypted Tablespace

Refer to "5.5 Checking an Encrypted Tablespace".

6.6 Managing the Keystore

Describes how to manage master encryption keys when a key management system is used.

6.6.1 Changing the Master Encryption Key

To change the master encryption key, run the pgx_declare_external_master_key function as you originally set it. Refer to "6.2 Setting the Master Encryption Key" for more information.

Also, by specifying a different key management system name than the key management system name specified during installation, you can change the key management system to be used. Refer to"6.6.5 Changes to the Key Management System" for details.

6.6.2 Enabling Automatic Opening of the Keystore

You can automatically open a keystore at instance startup without entering a passphrase by specifying all credentials, including those that should be kept secret, in the key management system connection information file. To enable automatic keystore opening, run the pgx_keystore command.

Example of storing obfuscated credentials in the file sslkeypasprase.ksc

```
> pgx_keystore -s -o sslkeypassphrase.ksc
Enter secret:
```

Specify obfuscated credentials in the key management system connection information file.

```
kmip    mykmipsvr    kmip.example.com    5696    cert    sslcert=postgres.crt    sslkey=postgres.key
sslrootcert=root.crt    sslkeypassphrase-obf=sslkeypassphrase.ksc
```

The key management system connection information file is valid only on the computer on which it was created.

To disable automatic keystore opening, delete the file containing obfuscated credentials for the private key specified in sslkeypasserase-obf and delete the sslkeypasserase-obf option in the key management system connection information file.

Refer to "pgx_keystore" in the Reference for information on pgx_keystore command.

Refer to "Appendix A Parameters" for information on the key management system connection information file.

6.6.3 Changing Credentials for Key Management Systems

If the credentials for the key management service change, you must also change the credentials that Fujitsu Enterprise Postgres uses to connect to the key management system.

You can change the credentials used by Fujitsu Enterprise Postgres using the pgx_open_keystore function. The new credentials are used to connect to the new key management system.

If you have a streaming replication configuration, you must change credentials on all replicas.

Refer to "B.2.1 pgx_open_keystore" for information on the pgx_open_keystore function.

6.6.4 Verifying the Master Encryption Key

The view pgx_tde_master_key shows information about your master encryption key. For more information about columns, Refer to "D.1 pgx_tde_master_key" for more information about columns..

6.6.5 Changes to the Key Management System

If you need to change the linked key management system after installing the transparent data encryption function that uses the key management system, you can do so by following the procedure below.

After this step, data in Fujitsu Enterprise Postgres will be encrypted using encryption keys on the new key management system.

Defining a new key management system

If the new key management system is not listed in the key management system's connection information file, add a definition to that configuration file. Give the old key management system and the new key management system different key management system names.

Reload the configuration file for the changes to take effect.

Declaring the master encryption key to use

Use the pgx_dexlare_external_master_key function to declare a new encryption key to use. Specify the name you gave the new key management system as the key management system name. Other arguments required are the key ID to use on the new key management system and credentials. Upon successful completion, it will be encrypted using the encryption key on the new key management system.

The master encryption key that encrypts the backup data before changing the key management system exists only on the old key management system. As such, you need the old key management system and the encryption keys residing there for any period of time when you might restore backup data from before the change. If you delete the encryption key on the old key management system, destroy the old

key management system, cancel the key management service, etc., you will not be able to decrypt the backup data and you will not be able to use the data.

6.7 Backing Up and Restoring/Recovering the Database

Fujitsu Enterprise Postgres enables you to use the five backup and recovery methods described below.

Backup and recovery using WebAdmin

- Backup

  WebAdmin backs up encrypted data.

- Recovery

  If you recover to a point in time when the old master encryption key was in use, change to the most recent master encryption key immediately after recovery.

  Enable automatic opening of the keystore in accordance with the procedure described in "6.6.2 Enabling Automatic Opening of the Keystore". Then, use WebAdmin to recover the database.

Backup and recovery using the pgx_dmpall and pgx_rcvall commands

- Backup

  The pgx_dmpall command backs up the encrypted data.

- Recovery

  If you recover to a point in time when the old master encryption key was in use, change to the most recent master encryption key immediately after recovery.

  Configure automatic opening of the key store as necessary.

  If automatic opening of the keystore is not enabled, execute the pgx_rcvall command with the --kms-secret option specified. This will display the prompt for the passphrase to be entered.

Dump and restore using SQL

- Backup

  The files output by the pg_dump and pg_dumpall commands are not encrypted. You should, therefore, encrypt the files using OpenSSL commands or other means before saving them, as described in "5.8 Importing and Exporting the Database" below.

- Restore

  If the backup data has been encrypted using, for example Open SSL commands, decrypt that data.

  The data generated by the pg_dumpall command includes a specification to encrypt tablespaces by default. For this reason, the psql command encrypts tablespaces during restoration.

File system level backup and restore

- Backup

  Stop the instance and backup the data directory and the tablespace directory using the file copy command of the operating system. The files of encrypted tablespaces are backed up in the encrypted state.

- Restore

  Stop the instance and use the OS file copy command to restore the data storage directory or tablespace directory.

  If you recover to a point in time when the old master encryption key was in use, change to the most recent master encryption key immediately after recovery.

Continuous archiving and point-in-time recovery

- Backup

  The pg_basebackup command backs up the encrypted data as is.

- Recovery

  If you recover to a point in time when the old master encryption key was in use, change to the most recent master encryption key immediately after recovery.

  Configure automatic opening of the key store as necessary.

  If automatic opening of the keystore is not enabled, execute the pg_ctl command to start the instance with the --kms-secret option specified. This will display the prompt for the passphrase to be entered.

See

- Refer to "pg_ctl" under "Reference" in the PostgreSQL Documentation for information on the pg_ctl command.
- Refer to "Reference" in the PostgreSQL Documentation for information on the following commands:
  - psql
  - pg_dump
  - pg_basebackup
- Refer to the Reference for information on the following commands:
  - pgx_rcvall
  - pgx_dmpall
  - pg_dumpall

6.8 Importing and Exporting the Database

Refer to "5.8 Importing and Exporting the Database".

6.9 Encrypting Existing Data

Refer to "5.9 Encrypting Existing Data".

6.10 Operations in Cluster Systems

This section describes how to use transparent data encryption on cluster systems such as high-availability systems, streaming replication, and database multiplexing.

6.10.1 HA Clusters that do not Use Database Multiplexing

Take the following points when using transparent data encryption with a key management system as a key store in an HA cluster environment that does not use database multiplexing.

Placement and automatic opening of the connection information file of the key management system

The file that describes the connection information of the key management system specified in the tde_kms.kms_conninfo_file parameter of the postgresql.conf file, and the files such as certificates that are referenced from that file can be shared by the primary server and the standby server. However, the obfuscated credential file used to enable automatic opening of the keystore must be created and placed on each server according to the instructions for enabling automatic opening.

If not shared, it must be possible to connect to the key management system used from the standby server with the same key management system name as the key management system name set on the primary server. Place the connection information file and files such as

certificates referenced from the connection information file on the standby server. The obfuscated credential file used to enable automatic opening of the keystore must be created and placed on each server according to the instructions in enabling automatic opening.

Changing credentials for key management systems

If the credentials for the key management system have changed, use the pgx_open_keystore function on the primary server to change the credentials. Re-enable automatic opening of the keystore on the standby server.

See

Refer to the Cluster Operation Guide (PRIMECLUSTER) for information on building a cluster system environment for performing failover operations linked to cluster software.

6.10.2 Database Multiplexing Mode

Note the following when using transparent data encryption with a key management system as a key store in environments that use streaming replication, or database multiplexing with streaming replication.

Placement and automatic opening of the connection information file of the key management system

The file that describes the connection information of the key management system specified in the tde_kms.kms_conninfo_file parameter of the postgresql.conf file, and the files such as certificates that are referenced from that file can be shared by the primary server and the standby server. However, the obfuscated credential file used to enable automatic opening of the keystore must be created and placed on each server according to the instructions for enabling automatic opening.

If not shared, it must be possible to connect to the key management system to be used with the same key management system name as the key management system name set on the primary server. Place the connection information file and files such as certificates referenced from the connection information file on all servers that configure database multiplexing mode. The obfuscated credential file used to enable automatic opening of the keystore must be created and placed on each server according to the instructions in enabling automatic opening.

Changing credentials for key management systems

If the credentials for the key management system are changed, use the pgx_open_keystore function on all servers that configure database multiplexing to change the credentials.

Starting a standby server

Open the keystore when starting the standby server. This is required to decrypt and replay the encrypted WAL received from the primary server. To open a keystore, use the pg_ctl or pgx_rcvall command with --kms-secret and provide your credentials, or enable automatic opening of the keystore.

Changing the master encryption key

Change the master encryption key on the primary server. No need to restart the standby server or reopen the keystore. Changes to the master encryption key are also reflected on the standby server.

See

Refer to "pgx_rcvall " in the Reference for information on pgx_rcvall command.

Refer to "pg_ctl" under "Reference" in the PostgreSQL Documentation for information on pg_ctl command.

Refer to "High Availability, Load Balancing, and Replication" under "Server Administration" in the PostgreSQL Documentation for information on how to set up streaming replication.

6.11 Security-Related Notes

Refer to "5.11 Security-Related Notes**".**

6.12 Tips for Installing Built Applications

Refer to "5.12 Tips for Installing Built Applications**".**

Chapter 7 Data Masking

Data masking is a feature that can change the returned data for queries generated by applications, so that it can be referenced by users. For example, for a query of employee data, digits except the last four digits of an eight-digit employee number can be changed to "*" so that it can be used for reference.

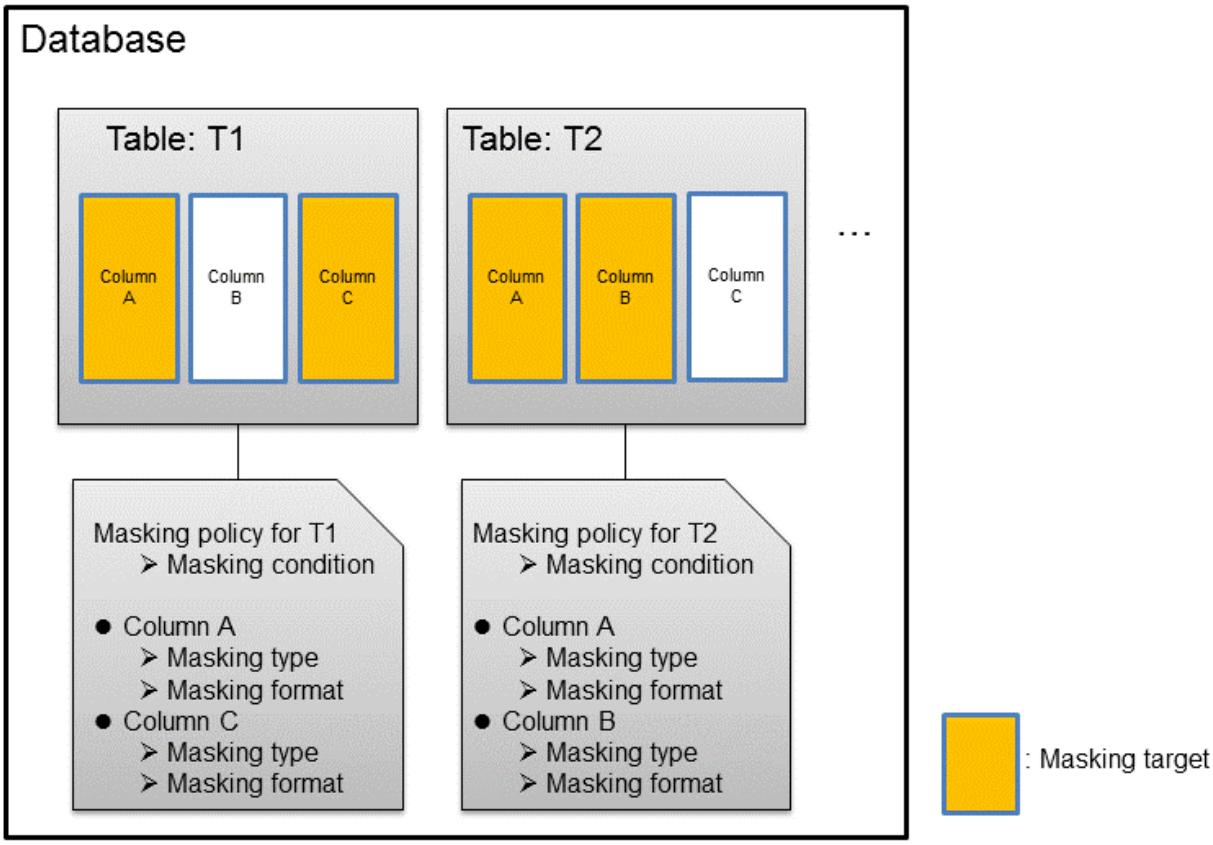
Note

When using this feature, it is recommended that the changed data be transferred to another medium for users to reference. This is because, if users directly access the database to extract the masked data, there is a possibility that they can deduce the original data by analyzing the masking policy or query result to the masking target column.

7.1 Masking Policy

Masking policy is a method of changing data under specific conditions when it is returned for a query from an application. One masking policy can be created per table. You can configure masking target, masking type, masking condition and masking format in a masking policy.

Figure 7.1 Masking policy



Note

When a masking policy is defined, the search performance for the corresponding table may deteriorate.

7.1.1 Masking Target

Masking target refers to a column to which a masking policy will be applied. When referring to a masking target or a function that includes a masking target, the execution result will be changed and obtained.

The following commands can change the execution result:

- SELECT
- COPY
- pg_dump
- pg_dumpall

- If a masking target is specified to INSERT...SELECT target columns, processing will be performed using data before change.
- If a masking target other than SELECT target columns is specified, processing will be performed using data before change.
- If a masking target is specified in a function where the data type will be converted, an error will occur.

7.1.2 Masking Type

Masking type is a method to change column data that is returned from queries. Specify the masking type in the function_type parameter. The following masking types can be specified and selected depending on the masking target data type.

Full masking

All the data in the specified column is changed. The changed value returned to the application that made the query varies depending on the column data type.
For example, 0 is used for a numeric type column and a space is used for a character type column.

Partial masking

The data in the specified column is partially changed.
For example, digits except the last four digits of an employee number can be changed to "*".

Regular expression masking

The data in the specified column is changed via a search that uses a regular expression.
For example, for strings such as email address that can have variable length, "*" can be used to change characters preceding "@" by using a regular expression. Regular expression masking can only be used for character type data.

- If multiple valid masking targets are specified for a function, the masking type for the left-most masking target will be applied.
  For example, if "SELECT GREATEST(c1, c2) FROM t1" is executed for numeric type masking target c1 and c2, the masking type for c1 will be applied.
- When masking the data that includes multibyte characters, do not specify partial masking for masking type. The result may not be as expected.

7.1.3 Masking Condition

Masking condition refers to the conditions configured to perform masking. Specify the masking condition in the expression parameter. Changed or actual data can be displayed for different users by defining masking condition. An expression that returns a boolean type result needs to be specified in masking condition and masking is performed only when TRUE is returned. Refer to "Value Expressions" in the PostgreSQL Documentation for information on the expressions that can be specified. Note that expressions that include a column cannot

be specified.
For example, when masking data only for "postgres" users, specify 'current_user = "postgres"' in the masking condition.

Information

Specify '1=1' so the masking condition is always evaluated to be TRUE and masking is performed all the time.

7.1.4 Masking Format

Masking format is a combination of change method and displayed characters when the masking condition is met. Masking format varies depending on the masking type. The following describes the masking format.

Full masking

With full masking, all characters are changed to values as determined by the database. Changed characters can be referenced in the pgx_confidential_values table. Also, replacement characters can be changed using the pgx_update_confidential_values system management function.

See

Refer to "7.3 Data Types for Masking" for information on the data types for which data masking can be performed.

Partial masking

With partial masking, data is changed according to the content in the function_parameters parameter. The method of specifying function_parameters varies depending on the data type.

Category	Method of specifying function_parameters
Numeric type	'*replacementCharacter*, *startPosition*, *endPosition*' - *replacementCharacter*: Specify the number to display. Specify a value from 0 to 9. - *startPosition*: Specify the start position of masking. Specify a positive integer. - *endPosition*: Specify the end position of masking. Specify a positive integer that is greater than *startPosition*. **Example** Specify as below to change the values from the 1st to 5th digits to 9. function_parameters := '9, 1, 5' In this example, if the original data is "123456789", it will be changed to "999996789".
Character type	*'inputFormat*, *outputFormat*, *replacementCharacter*, *startPosition*, *endPosition'* - *inputFormat*: Specify the current format of the data. Specify "V" for characters that will potentially be masked, and specify "F" for values such as spaces or hyphens that will not be masked. - *outputFormat*: Define the method to format the displayed data. Specify "V" for characters that will potentially be masked. Any character to be output can be specified for each character "F" in *inputFormat*. If you want to output a single quotation mark, specify two of them consecutively. - *replacementCharacter*: Specify any single character. If you want to output a single quotation mark, specify two of them consecutively. - *startPosition*: Specify the position of "V" as the start position of masking. For example, to specify the position of the 4th "V" from the left, specify 4. Specify a positive integer.

Category	Method of specifying function_parameters
	- *endPosition*: Specify the position of "V" as an end position of masking. When working out the end position, do not include positions of "F". For example, to specify the position of the 11th "V" from the left, specify 11. Specify a positive integer that is greater than *startPosition*. **Example** Specify as below to mask a telephone number other than the first three digits using *. function_parameters := 'VVVFVVVVFVVVV, VVV-VVVV-VVVV, *, 4, 11' In this example, if the original data is "012-3156-7890", it will be changed to "012-****-****".
Date/timestamp type	'MDYHMS' - M: Masks month. To mask month, enter the month from 1 to 12 after a lowercase letter m. Specify an uppercase letter M to not mask month. - D: Masks date. To mask date, enter the date from 1 to 31 after a lowercase letter d. If a value bigger than the last day of the month is entered, the last day of the month will be displayed. Specify an uppercase letter D to not mask date. - Y: Masks year. To mask year, enter the year from 1 to 9999 after a lowercase letter y. Specify an uppercase letter Y to not mask year. - H: Masks hour. To mask hour, enter the hour from 0 to 23 after a lowercase letter h. Specify an uppercase letter H to not mask hour. - M: Masks minute. To mask minute, enter the minute from 0 to 59 after a lowercase letter m. Specify an uppercase letter M to not mask minute. - S: Masks second. To mask second, enter the second from 0 to 59 after a lowercase letter s. Specify an uppercase letter S to not mask second. **Example** Specify as below to mask hour, minute, and second and display 00:00:00. function_parameters := 'MDYh0m0s0' In this example, if the original data is "2022-02-10 10:10:10", it will be changed to "2022-02-10 00:00:00".

See

- Refer to "B.3.2 pgx_create_confidential_policy" for information on function_parameters.
- Refer to "7.3 Data Types for Masking" for information on the data types for which masking can be performed.

Regular expression masking

With regular expression masking, data is changed according to the content of the regexp_pattern, regexp_replacement and regexp_flags parameters. For regexp_pattern, specify the search pattern using a regular expression. For regexp_replacement, specify the replacement character to use when data matches the search pattern. For regexp_flags, specify the regular expression flags.

Example

Specify as below to change all three characters starting from b to X.

regexp_pattern := 'b..'

regexp_replacement:= 'X'

regexp_flags := 'g'

In this example, if the original data is "foobarbaz", it will be changed to "fooXX".

See

- Refer to "POSIX Regular Expressions" in the PostgreSQL Documentation and check pattern, replacement, and flags for information on the values that can be specified for regexp_pattern, regexp_replacement, and regexp_flags.
- Refer to "7.3 Data Types for Masking" for information on the data types for which masking can be performed.

Note

- When column data type is character(*n*) or char(*n*) and if the string length after change exceeds n, the extra characters will be truncated and only characters up to the nth character will be displayed.
- When column data type is character varying(*n*) or varchar(*n*) and if the string length after change exceeds the length before the change, the extra characters will be truncated and only characters up to the length before change will be displayed.

7.2 Usage Method

Preparation

The following preparation is required to use this feature.

1. Set the postgresql.conf file parameters.

   Prepend "pgx_datamasking" to the shared_preload_libraries parameter.

2. Restart the instance.

3. Execute CREATE EXTENSION for the database that will use this feature.

   The target database is described as "postgres" here.

   Use the psql command to connect to the "postgres" database.

   #### Example

   ```
   postgres=# CREATE EXTENSION pgx_datamasking;
   CREATE EXTENSION
   ```

   #### Note

   You must always prepend "pgx_datamasking" to the "shared_preload_libraries" parameter.

   #### Information

   - Specify "false" for pgx_datamasking.enable to not use this feature. Data will not be masked even if a masking policy is configured. This feature becomes available again once "true" is specified for pgx_datamasking.enable. This setting can be made

by specifying a SET statement or specifying a parameter in the postgresql.conf file.
Example

```
postgres=# SET pgx_datamasking.enable=false;
```

- Hereafter, also perform this preparatory task for the "template1" database, so that this feature can be used by default when creating a new database.

Usage

To perform masking, a masking policy needs to be configured. The masking policy can be created, changed, confirmed, enabled, disabled or deleted during operation.
The procedures to perform these tasks are explained below with examples.

1. Creating a masking policy
2. Changing a masking policy
3. Confirming a masking policy
4. Enabling and disabling a masking policy
5. Deleting a masking policy

Note

Only database superusers can configure masking policies.

7.2.1 Creating a Masking Policy

An example of the operation on the server is shown below.

1. Create a masking policy
   Execute the pgx_create_confidential_policy system management function to create a masking policy.
   The following values are configured in this example.
   - Masking target: Numeric type c1
   - Masking type: FULL
   - Masking condition: '1=1'

```
postgres=# select pgx_create_confidential_policy(table_name := 't1', policy_name := 'p1',
expression := '1=1', column_name := 'c1', function_type := 'FULL');
 pgx_create_confidential_policy
--------------------------------
 t
(1 row)
```

2. Confirm the displayed data
   Confirm that the masking target data (column c1) has been correctly changed.

```
postgres=# select * from t1;
 c1 |      c2
----+---------------
  0 | 012-3456-7890
  0 | 012-3456-7891
  0 | 012-3456-7892
(3 row)
```

See

- Refer to "B.3.2 pgx_create_confidential_policy" for information on the pgx_create_confidential_policy system management function.

Note

- Only one masking policy can be created per table.
- All users can view the masking policy created, so do not grant the login privilege of the database where this feature is set to the users who refer to the changed data. Masking policies are defined in the "pgx_confidential_columns", "pgx_confidential_policies" and "pgx_confidential_values" tables.

7.2.2 Changing a Masking Policy

1. An example of the operation on the server is shown below.
2. Change a masking policy
   Execute the pgx_alter_confidential_policy system management function to change a masking policy.
   The following values are changed in this example.
   - Content of change: Add a masking target
   - Masking target: Character type c2
   - Masking type: PARTIAL
   - Masking condition: 'VVVFVVVVFVVVV, VVV-VVVV-VVVV, *, 4, 11'

```
postgres=# select pgx_alter_confidential_policy(table_name := 't1', policy_name := 'p1',
action := 'ADD_COLUMN', column_name := 'c2', function_type := 'PARTIAL', function_parameters :=
'VVVFVVVVFVVVV, VVV-VVVV-VVVV, *, 4, 11');
 pgx_alter_confidential_policy
-------------------------------
 t
(1 row)
```

3. Confirm the displayed data
   Confirm that the masking target data has been correctly changed.

```
postgres=# select * from t1;
 c1 |      c2
----+--------------
  0 | 012-****-****
  0 | 012-****-****
  0 | 012-****-****
(3 row)
```

See

- Refer to "B.3.1 pgx_alter_confidential_policy" for information on the pgx_alter_confidential_policy system management function.

7.2.3 Confirming a Masking Policy

An example of the operation on the server is shown below.

1. Confirm information about a masking target where a masking policy is set
   Refer to the pgx_confidential_columns table to confirm the masking target where the masking policy is set.

```
postgres=# select * from pgx_confidential_columns;
 schema_name | table_name | policy_name | column_name | function_type |
function_parameters           | regexp_pattern | regexp_replacement | regexp_flags |
column_description
-------------+------------+-------------+-------------+---------------
+---------------------------------------+----------------+--------------------+-------------
+--------------------
 public      | t1         | p1          | c1          | FULL
|                                       |                |                    |             |
```

```
 public      | t1         | p1          | c2          | PARTIAL       | VVVFVVVVFVVVV, VVV-VVVV-
VVVV, *, 4, 11 |                |                    |             |
(2 row)
```

2. Confirm information about the masking policy content
   Refer to pgx_confidential_policies to confirm the masking policy content.

```
postgres=# select * from pgx_confidential_policies;
 schema_name | table_name | policy_name | expression | enable | policy_description
-------------+------------+-------------+------------+--------+--------------------
 public      | t1         | p1          | 1=1        | t      |
(1 row)
```

See

- Refer to "E.1 pgx_confidential_columns" for information on the pgx_confidential_columns table.
- Refer to "E.2 pgx_confidential_policies" for information on the pgx_confidential_policies table.

7.2.4 Enabling and Disabling a Masking Policy

An example of the operation on the server is shown below.

1. Disable a masking policy
   Execute the pgx_enable_confidential_policy system management function to disable a masking policy.

```
postgres=# select pgx_enable_confidential_policy(table_name := 't1', policy_name := 'p1',
enable := 'f');
 pgx_enable_confidential_policy
--------------------------------
 t
(1 row)
```

2. Confirm the displayed data
   Confirm that the original data is displayed by disabling the masking policy.

```
postgres=# select * from t1;
 c1 |      c2
----+---------------
  1 | 012-3456-7890
  2 | 012-3456-7891
  3 | 012-3456-7892
(3 row)
```

3. Enable a masking policy
   Execute the pgx_enable_confidential_policy system management function to enable a masking policy.

```
postgres=# select pgx_enable_confidential_policy(table_name := 't1', policy_name := 'p1',
enable := 't');
 pgx_enable_confidential_policy
--------------------------------
 t
(1 row)
```

4. Confirm the displayed data
   Confirm that the masking target data has been correctly changed.

```
postgres=# select * from t1;
 c1 |      c2
----+---------------
  0 | 012-****-****
  0 | 012-****-****
```

```
  0 | 012-****-****
(3 row)
```

See

- Refer to "B.3.4 pgx_enable_confidential_policy" for information on the pgx_enable_confidential_policy system management function.

7.2.5 Deleting a Masking Policy

An example of the operation on the server is shown below.

1. Delete a masking policy
   Execute the pgx_drop_confidential_policy system management function to delete a masking policy.

```
postgres=# select pgx_drop_confidential_policy(table_name := 't1', policy_name := 'p1');
 pgx_drop_confidential_policy
------------------------------
 t
(1 row)
```

2. Confirm the displayed data
   Confirm that the original data is displayed by deleting the masking policy.

```
postgres=# select * from t1;
 c1 |      c2
----+--------------
  1 | 012-3456-7890
  2 | 012-3456-7891
  3 | 012-3456-7892
(3 row)
```

See

- Refer to "B.3.3 pgx_drop_confidential_policy" for information on the gx_drop_confidential_policy function.

7.3 Data Types for Masking

The data types for which data masking can be performed are shown below.

Category	Data type	Masking type		
		Full masking	Partial masking	Regular expression masking
Numeric type	smallint	Y	Y	N
	integer	Y	Y	N
	bigint	Y	Y	N
	decimal	Y	Y	N
	numeric	Y	Y	N
	float	Y	Y	N
	real	Y	Y	N
	double precision	Y	Y	N

Category	Data type	Masking type		
		Full masking	Partial masking	Regular expression masking
Character type	character varying(*n*)	Y	Y	Y
	varchar(*n*)	Y	Y	Y
	character(*n*)	Y	Y	Y
	char(*n*)	Y	Y	Y
Date/timestamp type	date	Y	Y	N
	timestamp	Y	Y	N

Note

Even if the data type can be masking, if the data is a special value (NaN, Infinity, -Infinity), it is not.

7.4 Security Notes

- Starting with FEP 10, logical replication is available, which allows non-backed up clusters to subscribe to databases where data masking policies are enabled. Logical replication allows publisher and subscriber databases to have their own or the same data masking policies.

  In this scenario, the user must disable data masking on the publisher database whenever a subscription is created. This ensures that subscribers are able to obtain the original data (initial copy) instead of the masked version. Then, it is the user's responsibility to set masking policies to each subscribed database.

- Take strong caution in publishing data masking's confidential tables (pgx_confidential_policies, pgx_confidential_columns, etc.) unless the user is publishing all tables of the database and wants to apply the same data masking's policies on the subscribed database for all of them.

  Otherwise, as these confidential tables contain the masking policies for all tables of the database, confidential policies of unpublished tables may be unintentionally published. Additionally, it is not possible to apply different data masking policies on the subscriber database.

Chapter 8 Periodic Operations

This chapter describes the operations that must be performed periodically when running daily database jobs.

8.1 Configuring and Monitoring the Log

Fujitsu Enterprise Postgres enables you to output database errors and warnings to a log file.

This information is useful for identifying if errors have occurred and the causes of those errors.

By default, this information is output to the system log. It is recommended that you configure Fujitsu Enterprise Postgres to collect logs from its log files (for example, log_destination) before operating Fujitsu Enterprise Postgres.

Periodically monitor the log files to check if any errors have occurred.

See

- Refer to "Error Reporting and Logging" under "Server Administration" in the PostgreSQL Documentation for information on logs.
- Refer to "Configuring Parameters" in the Installation and Setup Guide for Server for information on log settings when operating with WebAdmin.

8.2 Monitoring Disk Usage and Securing Free Space

When a database is used for an extended period, free space on the disk is continuously consumed and in some cases the disk space runs out. When this happens, database jobs may stop and no longer run.

You should, therefore, periodically monitor the usage of disk space, and delete obsolete files located in the disk.

Monitor the disk usage of the disk where the following directories are located:

- Data storage destination directory
- Transaction log storage destination (if the transaction log is stored in a different directory from the data storage destination directory)
- Backup data storage destination directory
- Tablespace storage destination directory

8.2.1 Monitoring Disk Usage

To check the disk usage, use the following operating system commands:

- df command

You can even use SQL statements to check tables and indexes individually.

Refer to "Determining Disk Usage" under "Server Administration" in the PostgreSQL Documentation for information on this method.

Information

If you are using WebAdmin for operations, a warning is displayed when disk usage reaches 80%

8.2.2 Securing Free Disk Space

Secure free disk space by using the following operating system commands to delete unnecessary files, other than the database, from the same disk unit.

- rm command

You can also secure disk space by performing the following tasks periodically:

- To secure space on the data storage destination disk:

  Execute the REINDEX statement. Refer to "8.5 Reorganizing Indexes" for details.

- To secure space on the backup data storage destination disk:

  Execute backup using WebAdmin or the pgx_dmpall command.

8.3 Automatically Closing Connections

If an application stops responding and abnormally terminates for any reason, the connection from the application may remain active on the database server. If this situation continues for an extended period, other applications attempting to connect to the database server may encounter an error, or an error indicating that the tables are unavailable may occur.

It is, therefore, recommended that idle connections be closed automatically at regular intervals.

Set the following parameters in the postgresql.conf file to indicate the time permitted to elapse before a connection is closed.

Parameter name	Setting	Description
tcp_keepalives_idle	Time until keepalive is sent (seconds) If 0, the default value of the system is used.	Sends keepalive to an idle connection at the specified interval in seconds It is recommended to specify 30 seconds.
tcp_keepalives_interval	keepalive send interval (seconds) If 0, the default value of the system is used.	Sends keepalive at the specified interval It is recommended to specify 10 seconds.
tcp_user_timeout	Time to wait for a response from the server (milliseconds) If 0, the default value of the system is used. If not set, the behavior is the same as if 0 were specified.	After establishing the connection, when sending from the client to the server, if the TCP resend process operates, specify the time until it is considered to be disconnected. If a value other than 0 is specified in this parameter, the time until automatic disconnection is determined by the waiting time specified in this parameter. The actual wait time is until the timing of the first keepalive retransmission after the time specified by this parameter has elapsed.

Note

If a value other than 0 is specified for the tcp_user_timeout parameter, the waiting time set by the tcp_keepalives_idle parameter and tcp_keepalives_interval parameter will be invalid and the waiting time specified by the tcp_user_timeout parameter will be used.

See

Refer to "Connection Settings" under "Server Administration" in the PostgreSQL Documentation for information on the parameters.

8.4 Monitoring the Connection State of an Application

Fujitsu Enterprise Postgres does not immediately delete the updated or deleted data. If the VACUUM determines there are no transactions that reference the database, Fujitsu Enterprise Postgres collects obsolete data.

However, obsolete data is not collected if there are connections that have remained active for an extended period or connections occupying resources. In this case the database may expand, causing performance degradation.

See

Refer to "Routine Vacuuming" under "Server Administration" in the PostgreSQL Documentation for information on the VACUUM command.

In such cases, you can minimize performance degradation of the database by monitoring problematic connections.

The following methods are supported for monitoring connections that have been in the waiting status for an extended period:

- 8.4.1 Using the View (pg_stat_activity)
- 8.4.2 Using pgAdmin

8.4.1 Using the View (pg_stat_activity)

Use the view (pg_stat_activity) to identify and monitor connections where the client has been in the waiting status for an extended period.

Example

The example below shows connections where the client has been in the waiting status for at least 60 minutes.

However, when considering continued compatibility of applications, do not reference system catalogs directly in the following SQL statements.

```
postgres=# select * from pg_stat_activity where backend_type = 'client backend' and state='idle in
transaction' and current_timestamp > cast(query_start + interval '60 minutes' as timestamp);
-[ RECORD 1 ]----+------------------------------
datid            | 13003
datname          | db01
pid              | 4638
leader_pid       |
usesysid         | 10
usename          | fsep
application_name | apl01
client_addr      | 192.33.44.15
client_hostname  |
client_port      | 27500
backend_start    | 2022-02-24 09:09:21.730641+09
xact_start       | 2022-02-24 09:09:23.858727+09
query_start      | 2022-02-24 09:09:23.858727+09
state_change     | 2022-02-24 09:09:23.858834+09
wait_event_type  | Client
wait_event       | ClientRead
state            | idle in transaction
backend_xid      |
backend_xmin     |
query_id         |
query            | begin;
backend_type     | client backend
```

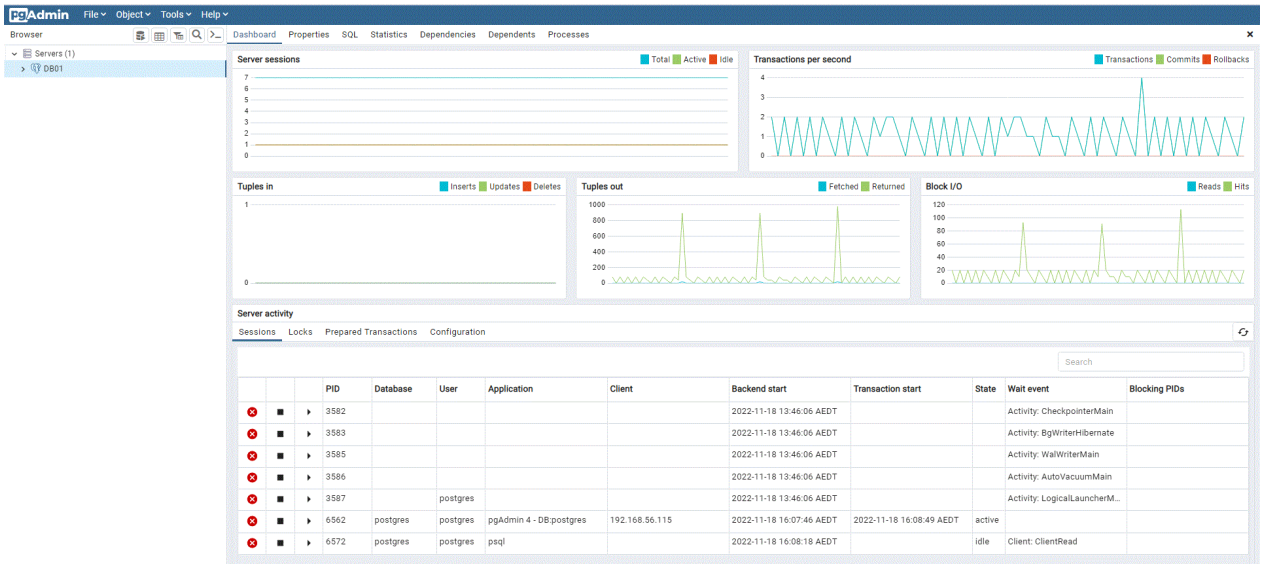
See

- Refer to "Notes on Application Compatibility" in the Application Development Guide for information on maintaining application compatibility.
- Refer to "The Statistics Collector" under "Server Administration" in the PostgreSQL Documentation for information on pg_stat_activity.

8.4.2 Using pgAdmin

This section describes the procedure for monitoring connections using [Server Status] in pgAdmin.

1. In the [Browser] pane, click the database server for monitoring.
2. In the [Dashboard] tab, identify client connections that have been in the waiting state for an extended period.



8.5 Reorganizing Indexes

Normally, a database defines indexes in tables, but if data is frequently updated, indexes can no longer use free space in the disk efficiently. This situation can also cause a gradual decline in database access performance.

To rearrange used space on the disk and prevent the database access performance from declining, it is recommended that you periodically execute the REINDEX command to reorganize indexes.

Check the disk usage of the data storage destination using the method described in "8.2 Monitoring Disk Usage and Securing Free Space".

Note

Because the REINDEX command retrieves the exclusive lock for an index being processed and locks writing of tables that are the source of the index, other processes that access these may stop while waiting to be locked.

Therefore, it is necessary to consider measures such as executing the command after the task is completed.

See

Refer to "Routine Reindexing" under "Server Administration" in the PostgreSQL Documentation for information on reorganizing indexes by periodically executing the REINDEX command.

Point

Typically, reorganize indexes once a month at a suitable time such as when conducting database maintenance. Use SQL statements to check index usage. If this usage is increasing on a daily basis, adjust the frequency of recreating the index as compared to the free disk space.

The following example shows the SQL statements and the output.

However, when considering continued compatibility of applications, do not reference system catalogs and functions directly in the following SQL statements. Refer to "Notes on Application Compatibility" in the Application Development Guide for details.

[SQL statements]

```
SELECT
  nspname AS schema_name,
  relname AS index_name,
  round(100 * pg_relation_size(indexrelid) / pg_relation_size(indrelid)) / 100 AS index_ratio,
  pg_size_pretty(pg_relation_size(indexrelid)) AS index_size,
  pg_size_pretty(pg_relation_size(indrelid)) AS table_size
FROM pg_index I
  LEFT JOIN pg_class C ON (C.oid = I.indexrelid)
  LEFT JOIN pg_namespace N ON (N.oid = C.relnamespace)
WHERE
  C.relkind = 'i' AND
  pg_relation_size(indrelid) > 0
ORDER BY pg_relation_size(indexrelid) DESC, index_ratio DESC;
```

[Output]

```
 schema_name |            index_name            | index_ratio | index_size | table_size
-------------+----------------------------------+-------------+------------+------------
 public      | pgbench_accounts_pkey            |        0.16 | 2208 KB    | 13 MB
 pg_catalog  | pg_depend_depender_index         |         0.6 | 224 KB     | 368 KB
 pg_catalog  | pg_depend_reference_index        |        0.58 | 216 KB     | 368 KB
...
```

See

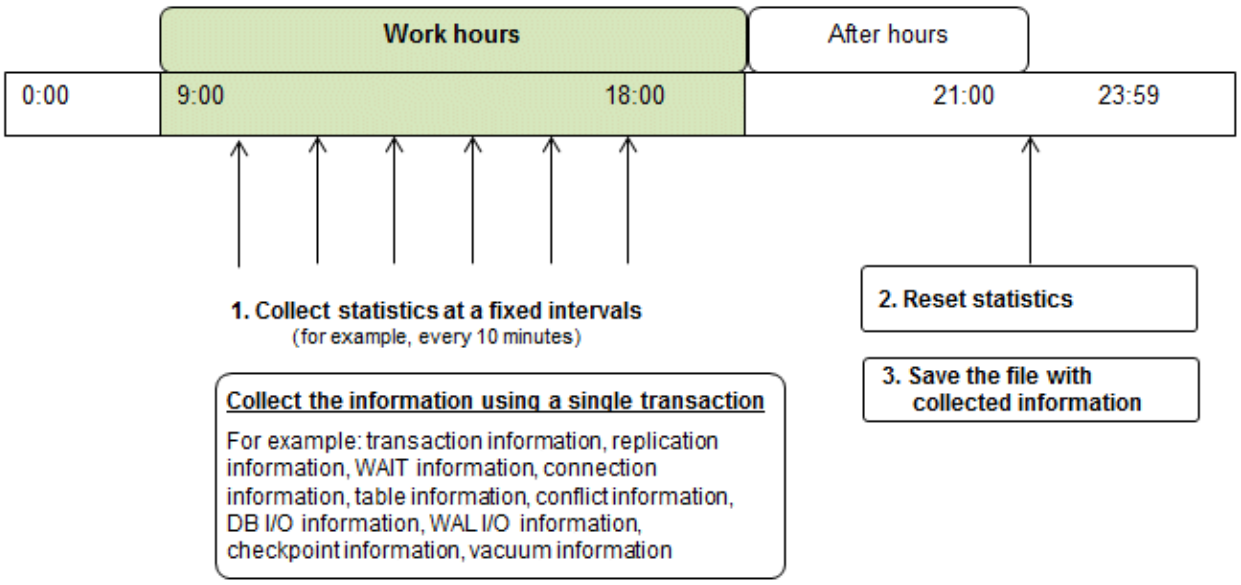
Refer to "Notes on Application Compatibility" in the Application Development Guide for information on maintaining application compatibility.

8.6 Monitoring Database Activity

Fujitsu Enterprise Postgres enables you to collect information related to database activity. By monitoring this information, you can check changes in the database status.

This information includes wait information for resources such as internal locks, and is useful for detecting performance bottlenecks. Furthermore, you should collect this information in case you need to request Fujitsu technical support for an investigation.

Figure 8.1 Overview of information collection



1. Collect statistics at fixed intervals during work hours.

   Accumulate the collected information into a file.

   Wherever possible, collect data from the various statistics views using a single transaction, because it enables you to take a snapshot of system performance at a given moment.

   Refer to "8.6.1 Information that can be Collected" for information on the system views that can be collected.

2. Reset statistics after work hours, that is, after jobs have finished.

   Refer to "8.6.3 Information Reset" for information on how to reset statistics.

3. Save the file with collected information.

   Keep the file with collected information for at least two days, in order to check daily changes in performance and to ensure that the information is not deleted until you have sent a query to Fujitsu technical support.

Where jobs run 24 hours a day, reset statistics and save the file with collected information when the workload is low, for example, at night.

Note

Statistics cumulatively add the daily database value, so if you do not reset them, the values will exceed the upper limit, and therefore will not provide accurate information.

The subsections below explain the following:

- Information that can be collected
- Collection configuration
- Information reset

8.6.1 Information that can be Collected

Information that can be collected is categorized into the following types:

- Information common to PostgreSQL
- Information added by Fujitsu Enterprise Postgres

Information common to PostgreSQL

See

Refer to "Monitoring Database Activity" under "Server Administration" in the PostgreSQL Documentation for information on information common to PostgreSQL.

Information added by Fujitsu Enterprise Postgres

You can collect the following information added by Fujitsu Enterprise Postgres.

Table 8.1 Information added by Fujitsu Enterprise Postgres

View name	Description
pgx_stat_lwlock	Displays statistic related to lightweight lock, with each type of content displayed on a separate line. This information helps to detect bottlenecks. Refer to "C.2 pgx_stat_lwlock" for details.
pgx_stat_latch	Displays statistics related latches, with each type of wait information within Fujitsu Enterprise Postgres displayed on a separate line. This information helps to detect bottlenecks. Refer to "C.3 pgx_stat_latch" for details.

View name	Description
pgx_stat_walwriter	Displays statistics related to WAL writing, in a single line. Refer to "C.4 pgx_stat_walwriter" for details.
pgx_stat_sql	Displays statistics related to SQL statement executions, with each type of SQL statement displayed on a separate line. Refer to "C.5 pgx_stat_sql" for details.
pgx_stat_gmc	Displays statistics related to Global Meta Cache hit ration and used memory size. Refer to "C.6 pgx_stat_gmc" for detail. Also refer to Chapter 13 Global Meta Cache" for information on the Global Meta Cache.

8.6.2 Collection Configuration

The procedure for configuring collection depends on the information content.

- Information common to PostgreSQL
- Information added by Fujitsu Enterprise Postgres

Information common to PostgreSQL

See

Refer to "The Statistics Collector" in "Monitoring Database Activity" under "Server Administration" in the PostgreSQL Documentation for information on information common to PostgreSQL.

Information added by Fujitsu Enterprise Postgres

Information added by Fujitsu Enterprise Postgres is collected by default.

To enable or disable information collection, change the configuration parameters in postgresql.conf. The following table lists the views for which you can enable or disable information collection, and the configuration parameters.

View name	Parameter
pgx_stat_lwlock pgx_stat_latch	track_waits (*1)
pgx_stat_sql	track_sql
pgx_stat_gmc	track_gmc

Remarks: You cannot change the collection status for pgx_stat_walwriter.

*1: When executing the SQL statement with EXPLAIN ANALYZE, processing time may increase because of this information collection. It is recommended to set this parameter to "off" when executing EXPLAIN ANALYZE to check the processing time.

Refer to "Appendix A Parameters" for information on the parameters.

8.6.3 Information Reset

This section describes how to reset information.

Information added by Fujitsu Enterprise Postgres

You can reset information added by Fujitsu Enterprise Postgres by using the pg_stat_reset_shared function in the same way as for information common to PostgreSQL.

Configure the following parameters in the pg_stat_reset_shared function:

Function	Type of return value	Description
pg_stat_reset_shared(text)	void	Reset some cluster-wide statistics counters to zero, depending on the argument (requires superuser privileges). Calling pg_stat_reset_shared('lwlock') will zero all counters shown in pgx_stat_lwlock. Similarly, in the following cases, all values of the pertinent statistics counter are reset: - If pg_stat_reset_shared('latch') is called: All values displayed in pgx_stat_latch - If pg_stat_reset_shared('walwriter') is called: All values displayed in pgx_stat_walwriter - If pg_stat_reset_shared('sql') is called: All values displayed in pgx_stat_sql - If pg_stat_reset_shared('gmc') is called: All values except size column in pgx_stat_gmc

See

Refer to "Statistics Functions" in "Monitoring Database Activity" under "Server Administration" in the PostgreSQL Documentation for information on other parameters of the pg_stat_reset_shared function.

Chapter 9 Streaming Replication Using WebAdmin

This chapter describes how to create a streaming replication cluster using WebAdmin.

Streaming replication allows the creation of one or more standby instances, which connect to the master instances and replicate the data using WAL records. The standby instance can be used for read-only operations.

WebAdmin can be used to create a streaming replication cluster. WebAdmin allows the creation of a cluster in the following configurations:

- Master-Standby Configuration: This configuration creates a master and standby instance together.
- Standby Only Configuration: This configuration creates a standby instance from an already existing instance.

Point

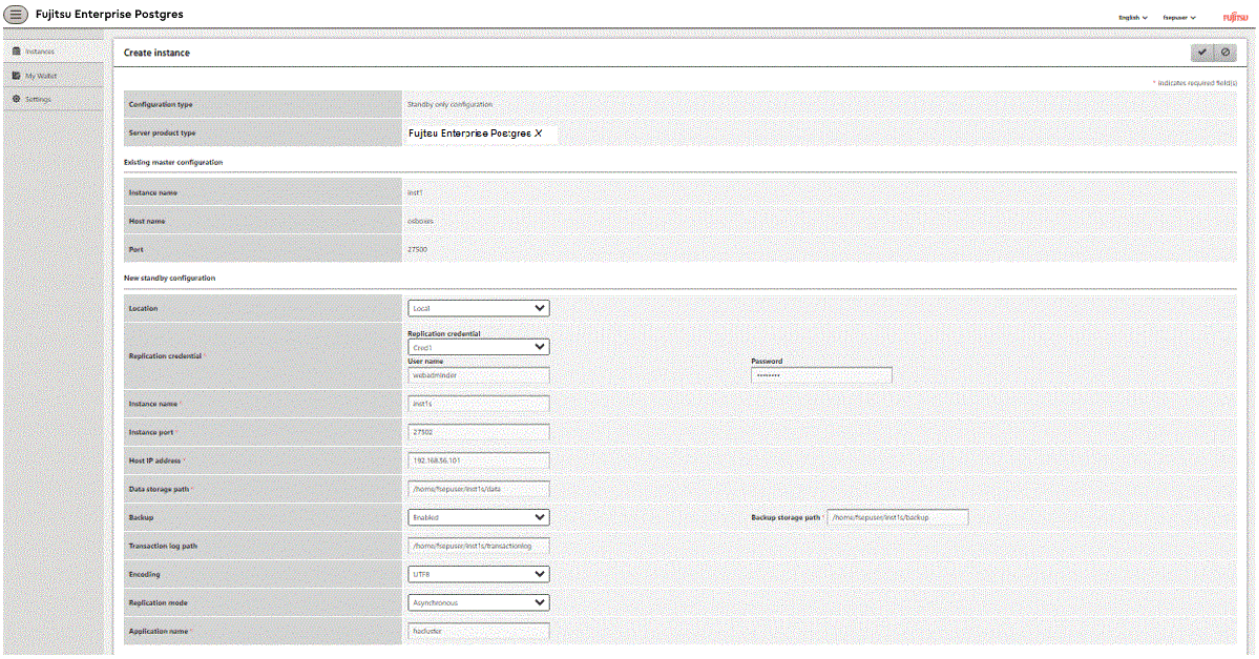
- A standby instance can be created from a standalone instance, a master instance, or even from another standby instance.
- If a streaming replication cluster is created using WebAdmin, the network with the host name (or IP address) specified in [Host name] will be used across sessions of WebAdmin, and also used as the log transfer network.
- To use a network other than the job network as the log transfer network, specify the host name other than the job network one in [Host name].

9.1 Creating a Standby Instance

Follow the procedure below to create a standby instance.

1. In the [Instances] tab, select the instance from which a standby instance is to be created.
2. Click .
3. Enter the information for the standby instance to be created. In the example below, a standby instance is created from instance "inst1".

   The instance name, host address and port of the selected instance are already displayed for easy reference.



Enter the following items:

- [Location]: Whether to create the instance in the server that the current user is logged in to, or in a remote server. The default is "Local", which will create the instance in the server machine where WebAdmin is currently running.

- [Replication credential]: The user name and password required for the standby instance to connect to the master instance. The user name and password can be entered or selected from the Wallet. Refer to "Appendix H WebAdmin Wallet" for information on creating wallet entries.
- [Instance name]: Name of the standby database instance to create.

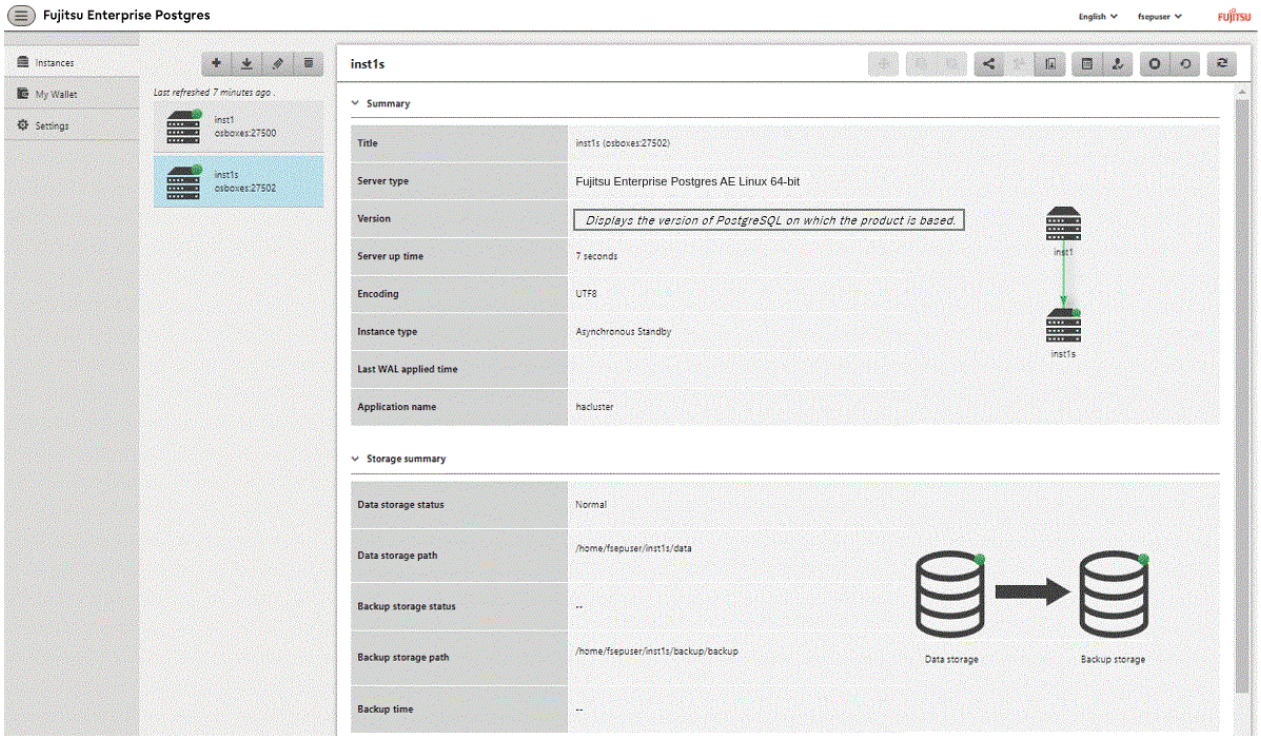
  The name must meet the conditions below:

  - Maximum of 16 characters
  - The first character must be an ASCII alphabetic character
  - The other characters must be ASCII alphanumeric characters
- [Instance port]: Port number of the standby database instance.
- [Host IP address]: The IP address of the server machine where the standby instance is to be created. This information is needed to configure the standby instance to be connected to the master.
- [Data storage path]: Directory where the database data will be stored
- [Backup storage path]: Directory where the database backup will be stored
- [Transaction log path]: Directory where the transaction log will be stored
- [Encoding]: Database encoding system
- [Replication mode]: Replication mode of the standby instance to be created ("Asynchronous" or "Synchronous")
- [Application name]: The reference name of the standby instance used to identify it to the master instance.

  The name must meet the conditions below:

  - Maximum of 16 characters
  - The first character must be an ASCII alphabetic character
  - The other characters must be ASCII alphanumeric characters

4. Click ✔ to create the standby instance.
5. Once the standby instance is created successfully, select standby instance in the [Instances] tab. The following page will be displayed:



Note

- Backups are not possible for standby instances in WebAdmin. As a result, and are disabled and no value is shown for [Backup storage status] and [Backup time].
- If using WebAdmin to manage Mirroring Controller, the message below may be output to the server log or system log in the standby instance. No action is required, as the instance is running normally.

```
ERROR:  pgx_rcvall failed
ERROR:  pgx_rcvall: backup of the database has not yet been performed, or an incorrect backup
storage directory was specified
```

- Replication credential (user name and password) should not contain hazardous characters. Refer to "Appendix I WebAdmin Disallow User Inputs Containing Hazardous Characters".

9.2 Promoting a Standby Instance

Streaming replication between a master and standby instance can be discontinued using WebAdmin.

Follow the procedure below to promote a standby instance to a standalone instance, thereby discontinuing the streaming replication.

1. In the [Instances] tab, select the standby instance that needs to be promoted.
2. Click .
3. Click [Yes] from the confirmation dialog box.

The standby instance will be promoted and will become a standalone instance, which is not part of a streaming replication cluster.

Once the standby instance is promoted to become a standalone instance, the backup storage status will be "Error". This is because no backups are available when the instance is newly promoted to a standalone instance. The status will be reset if a new backup is performed by clicking [Solution] or .

9.3 Converting an Asynchronous Replication to Synchronous

Streaming replication between a master and standby instance can be configured to be in Asynchronous or Synchronous mode. This mode can be changed even after the standby instance was successfully created.

Follow the procedure below to convert an Asynchronous standby instance to Synchronous.

1. In the [Instances] tab, select the master instance of the relevant cluster.
2. Click .
3. In the [Streaming replication] section, edit the value for [Synchronous standby names].
   - Add the "Application name" of the standby instance you want to be in Synchronous mode.
4. Click .
5. Select the master instance and click .
6. Select the standby instance. [Instance type] will now show the updated status.

Note

- Converting an Asynchronous standby instance to Synchronous can cause the master instance to queue the incoming transactions until the standby instance is ready. For this reason, it is recommended that this operation be performed during a scheduled maintenance period.
- When adding a synchronous standby instance, Fujitsu Enterprise Postgres will only keep the first entry in [Synchronous standby names] in synchronous state.

- To learn more about the differences between synchronous and asynchronous standby modes and their behavior, refer to "Streaming Replication" in "High Availability, Load Balancing, and Replication" in the PostgreSQL Documentation.

9.4 Converting a Synchronous Replication to Asynchronous

Streaming replication between a master and standby instance can be configured to be in Asynchronous or Synchronous Mode. This mode can be changed even after the standby instance was successfully created.

Follow the procedure below to convert a Synchronous standby instance to Asynchronous.

1. In the [Instances] tab, select the master instance of the relevant cluster.
2. Click .
3. In the [Streaming replication] section, edit the value for [Synchronous standby names].
   - Remove the "Application name" of the standby instance you want to be in Asynchronous mode.
4. Click .
5. Select the master instance and click .
6. Select the standby instance. [Instance type] will now show the updated status.

Note

To learn more about the differences between synchronous and asynchronous standby modes and their behavior, refer to "Streaming Replication" in "High Availability, Load Balancing, and Replication" in the PostgreSQL Documentation.

9.5 Joining a Replication Cluster

WebAdmin facilitates the joining of an old master of the cluster as a standby node.

1. In the [Instances] tab, select the remote instance (from where the new cluster node will stream WAL entries), and then click .
2. Configure the node to accept streaming requests from the new node.
3. In the [Instances] tab, select the new standby instance (which needs to be connected to the cluster), and then click .
4. Set [Replication host name] to the remote instance.
5. Enter [Replication credential].

   Specify the user name and password required for the standby instance to connect to the remote instance. The user name and password can be entered or selected from the Wallet. Refer to "Appendix H WebAdmin Wallet" for information on creating wallet entries. Replication credential (user name and password) should not contain hazardous characters. Refer to "Appendix I WebAdmin Disallow User Inputs Containing Hazardous Characters".
6. Enter [Host IP address].

   Specify the IP address of the node where the standby instance was created.
7. Click to open the [Join replication cluster] dialog box.

   For Fujitsu Enterprise Postgres 13 and earlier instances, select [Restart later] or [Restart now], and then click [Yes] to set up the standby instance.
   For Fujitsu Enterprise Postgres 14 or later, click [Yes] to set up the standby instance.
8. Upon successful completion, the confirmation dialog box will be displayed.
9. Click [Close] to return to the instance details window.

The instance will become a standby instance, and will be part of the streaming replication cluster. The replication diagram will display the relationship between the standby instance and the remote instance. The user can change the replication relationship of the remote instance from asynchronous to synchronous (and vice versa) using the [Configuration] window.

Chapter 10 Installing and Operating the In-memory Feature

The in-memory feature enables fast aggregation using Vertical Clustered Index (VCI) and memory-resident feature.

VCI has a data structure suitable for aggregation, and features parallel scan and disk compression, which enable faster aggregation through reduced disk I/O.

The memory-resident feature reduces disk I/O that occurs during aggregation. It consists of the preload feature that reads VCI data to memory in advance, and the stable buffer feature that suppresses VCI data eviction from memory. The stable buffer feature secures the proportion specified by parameter in the shared memory for VCI.

This chapter describes how to install and operate the in-memory feature.

Note

This feature can only be used in Advanced Edition.

10.1 Installing Vertical Clustered Index (VCI)

This section describes the installation of VCI.

1. Evaluating whether to Install VCI
2. Estimating Resources
3. Setting up

10.1.1 Evaluating whether to Install VCI

VCI uses available resources within the server to increase scan performance.

It speeds up processing in many situations, and can be more effective in the following situations:

- Single table processing
- Processing that handles many rows in the table
- Processing that handles some columns in the table
- Processing that performs very heavy aggregation such as simultaneous sum and average aggregation

VCI will not be used in the following cases, so it is necessary to determine its effectiveness in advance:

- The data type of the target table or column contains VCI restrictions.
- The SQL statement does not meet the VCI operating conditions
- VCI is determined to be slower based on cost estimation

Note

If performing operations that use VCI, the full_page_writes parameter setting in postgresql.conf must be enabled (on). For this reason, if this parameter is disabled (off), operations that use VCI return an error. In addition, to perform operations for tables that do not create a VCI when the full_page_writes parameter setting is temporarily disabled (off), do not create a VCI or perform operations to tables that created a VCI during that time.

See

- Refer to "10.1.4 Data that can Use VCI" for information on VCI restrictions.

- Refer to "Scan Using a Vertical Clustered Index (VCI)" - "Operating Conditions" in the Application Development Guide for information on VCI operating conditions.

10.1.2 Estimating Resources

Estimate resources before setting up VCI.

Select the aggregation that you want to speed up and identify the required column data. The additional resources below are required according to the number of columns.

- Memory

  Secure additional capacity required for the disk space for the column for which VCI is to be created.

- Disk

  Secure additional disks based on the disk space required for the column for which VCI is to be created, as VCI stores column data as well as existing table data on the disk. It is recommended to provide a separate disk in addition to the existing one, and specify it as the tablespace to avoid impact on any other jobs caused by I/O.

Information

The operations on VCI can continue even if the memory configured for VCI is insufficient by using VCI data on the disk.

See

Refer to "Estimating Memory Requirements" and "Estimating Database Disk Space Requirements" in the Installation and Setup Guide for Server for information on how to estimate required memory and disk space.

10.1.3 Setting up

This section describes how to set up VCI.

Setup flow

1. Setting Parameters
2. Installing the Extensions
3. Creating VCI
4. Confirming that VCI has been Created

10.1.3.1 Setting Parameters

Edit postgresql.conf to set the required parameters for VCI. After that, start or restart the instance.

The following table lists the parameters that need or are recommended to be configured in advance:

Parameter name	Setting value	Description	Required
shared_preload_libraries	Literal 'vci, pg_prewarm'	VCI and shared library to be preloaded at server start.	Y
session_preload_libraries	Literal 'vci, pg_prewarm'	VCI and shared library to be preloaded at connection start.	Y
reserve_buffer_ratio	Percentage of shared memory to be used for stable buffer table	Proportion of shared memory to be used for a stable buffer table.	N

Parameter name	Setting value	Description	Required
vci.control_max_workers	Number of background workers that manage VCI	Number of background workers that manage VCI. Add this value to max_worker_processes.	N
vci.max_parallel_degree	Maximum number of background workers used for parallel scan	Maximum number of background workers used for parallel scan. Add this value to max_worker_processes.	N

Example

```
shared_preload_libraries = 'vci, pg_prewarm'
session_preload_libraries = 'vci, pg_prewarm'
reserve_buffer_ratio = 20
vci.control_max_workers = 8
vci.max_parallel_degree = 4
max_worker_processes = 18 # Example: If the initial value was 6, 6 + 8 + 4 = 18
```

Note

An error occurs if you use VCI to start instances when procfs is not mounted. Ensure that procfs is mounted before starting instances.

See

- Refer to "Appendix A Parameters" for information on all parameters for VCI. Refer also to default value for each parameter and details such as specification range in the same chapter. Refer to "Server Configuration" under "Server Administration" in the PostgreSQL documentation for information on shared_preload_libraries, session_preload_libraries, and max_worker_processes.

10.1.3.2 Installing the Extensions

Execute CREATE EXTENSION to install the VCI and pg_prewarm extensions. Both extensions need to be installed for each database.

- Installing VCI

```
db01=# CREATE EXTENSION vci;
```

- Installing pg_prewarm

```
db01=# CREATE EXTENSION pg_prewarm;
```

Note

- Only superusers can install VCI extensions.
- VCI extensions can only be installed in public schema.
- Some operations cannot be performed for VCI extensions. Refer to "10.2.1 Commands that cannot be Used for VCI" for details.

10.1.3.3 Creating a VCI

Execute the CREATE INDEX statement with the "USING vci" clause to create a VCI for the desired columns and the "WITH (stable_buffer=true)" clause to enable the stable buffer feature.

To use a separate disk for the VCI, specify the TABLESPACE clause.

```
db01=# CREATE INDEX idx_vci ON table01 USING vci (col01, col02) WITH (stable_buffer=true);
```

Note

- Some table types cannot be specified on the ON clause of CREATE INDEX. Refer to "10.1.4.1 Relation Types" for details.
- Some data types cannot be specified on the column specification of CREATE INDEX. Refer to "10.1.4.2 Data Types" for details.
- Some operations cannot be performed for VCI. Refer to "10.2.1 Commands that cannot be Used for VCI" for details.
- The same column cannot be specified more than once on the column specification of CREATE INDEX.
- VCI cannot be created for table columns that belong to the template database.
- CREATE INDEX creates multiple views named vci_*10digitRelOid_5digitRelAttr_1charRelType* alongside VCI itself. These are called VCI internal relations. Do not update or delete them as they are used for VCI aggregation.
- All data for the specified column will be replaced in columnar format when VCI is created, so executing CREATE INDEX on an existing table with data inserted takes more time compared with a general index (B-tree). Jobs can continue while CREATE INDEX is running.
- When CREATE INDEX USING VCI is invoked on a partitioned table, the default behavior is to recurse to all partitions to ensure they all have matching indexes. Each partition is first checked to determine whether an equivalent index already exists, and if so, that index will become attached as a partition index to the index being created, which will become its parent index. If no matching index exists, a new index will be created and automatically attached; the name of the new index in each partition will be determined as if no index name had been specified in the command. If the ONLY option is specified, no recursion is done, and the index is marked invalid. (ALTER INDEX ... ATTACH PARTITION marks the index valid, once all partitions acquire matching indexes.) Note, however, that any partition that is created in the future using CREATE TABLE ... PARTITION OF will automatically have a matching index, regardless of whether ONLY is specified.
- Parallel index build is not supported on VCI indexes.

10.1.3.4 Confirming that the VCI has been Created

Execute the SELECT statement to reference the pg_indexes catalog, and confirm that the VCI was created for the target columns.

Example

```
db01=# SELECT indexdef FROM pg_indexes WHERE indexdef LIKE '%vci%';
                        indexdef
-----------------------------------------------------------
CREATE INDEX idx_vci ON table01 USING vci (col01, col02)
(1 row)
```

10.1.4 Data that can Use VCI

This section describes on which relation types and for which data types VCIs can be created.

10.1.4.1 Relation Types

VCIs cannot be created on some relation types.

The ON clause of CREATE INDEX described in "10.1.3.3 Creating a VCI" cannot specify relations on which VCIs cannot be created.

- Relations on which VCIs can be created
  - Normal tables
  - UNLOGGED TABLEs

- Relations on which VCIs cannot be created
  - Materialized views
  - Temporary tables
  - Views
  - Temporary views
  - Foreign tables

10.1.4.2 Data Types

VCIs cannot be created for some data types.

The column specification of CREATE INDEX described in "10.1.3.3 Creating a VCI" cannot specify a column with data type on which VCIs cannot be created.

Category	Data type	Supported by VCI?
Numeric	smallint	Y
	integer	Y
	bigint	Y
	decimal	Y
	numeric	Y
	real	Y
	double precision	Y
	serial	Y
	bigserial	Y
Monetary	money	Y
Character	varchar(*n*)	Y
	char(*n*)	Y
	nchar	Y
	nvarchar(*n*)	Y
	text	Y
Binary	bytea	Y
Date/time	timestamp	Y
	timestamp with time zone	Y
	date	Y
	time	Y
	time with time zone	Y
	interval	Y
Boolean	boolean	Y
Geometric	point	N
	line	N
	lseg	N
	box	N
	path	N

Category	Data type	Supported by VCI?
	polygon	N
	circle	N
Network address	cidr	N
	inet	N
	macaddr	N
	macaddr8	N
Bit string	bit(*n*)	Y
	bit varying(*n*)	Y
Text search	tsvector	N
	tsquery	N
UUID	uuid	Y
XML	xml	N
JSON	json	N
	jsonb	N
Range	int4range	N
	int8range	N
	numrange	N
	tsrange	N
	tstzrange	N
	daterange	N
Object identifier	oid	N
	regproc	N
	regprocedure	N
	regoper	N
	regoperator	N
	regclass	N
	regtype	N
	regconfig	N
	regdictionary	N
pg_lsn type	pg_lsn	N
Array type	-	N
User-defined type (Basic type, enumerated type, composite type, and range type)	-	N

10.2 Operating VCI

This section describes how to operate VCI.

10.2.1 Commands that cannot be Used for VCI

Some operations cannot be performed for VCI extensions and VCI itself.

This section describes SQL commands that cannot be executed for the VCI extensions and VCI itself, and client application commands.

SQL commands

- Operations that cannot be performed for the VCI extension

Command	Subcommand	Description
ALTER EXTENSION	UPDATE	The VCI extension cannot be specified. This operation is not required for VCI.
	SET SCHEMA	
	ADD	
	DROP	
CREATE EXTENSION	SCHEMA	The subcommands on the left cannot be performed if the VCI extension is specified. This operation is not required for VCI.

- Operations that cannot be performed on relations containing a VCI

Command	Subcommand	Description
ALTER INDEX	SET	The subcommands on the left cannot be performed if a VCI is specified. If the operation is required, delete the VCI using DROP INDEX, and re-create it using CREATE INDEX after completing the operation.
	SET TABLESPACE	
	ALL IN TABLESPACE	
ALTER OPERATOR CLASS	RENAME TO	The subcommands on the left cannot be performed if a VCI is specified. This operation is not supported in VCI.
	OWNER TO	
	SET SCHEMA	
ALTER OPERATOR FAMILY	ADD	
	DROP	
	RENAME TO	
	OWNER TO	
	SET SCHEMA	
ALTER TABLE	ALL IN TABLESPACE *name* [OWNED BY *roleName*] SET TABLESPACE *newTablespace*	A tablespace that contains a VCI cannot be specified. If the operation is required, delete the VCI using DROP INDEX, and re-create it using CREATE INDEX after completing the operation.
	DROP [COLUMN] [IF EXISTS] *colName* [RESTRICT | CASCADE]	A column that contains a VCI cannot be specified. If the operation is required, delete the VCI using DROP INDEX, and re-create it using CREATE INDEX after completing the operation.
	ALTER [COLUMN] *colName* [SET DATA] TYPE *dataType* [COLLATE *collation*] [USING *expr*]	
	CLUSTER ON *indexName*	A VCI cannot be specified. This operation is not supported in VCI.
	REPLICA IDENTITY {DEFAULT | USING INDEX *indexName* | FULL | NOTHING}	

Command	Subcommand	Description
CLUSTER	-	A table that contains a VCI and VCI cannot be specified. If the operation is required, delete the VCI using DROP INDEX, and re-create it using CREATE INDEX after completing the operation.
CREATE INDEX	UNIQUE	The subcommands on the left cannot be performed if a VCI is specified. This operation is not supported in VCI.
	CONCURRENTLY	
	[ASC | DESC]	
	[NULLS { FIRST | LAST }]	
	WITH	
	WHERE	
	INCLUDE	
CREATE OPERATOR CLASS	-	A VCI cannot be specified. This operation is not supported in VCI.
CREATE OPERATOR FAMILY	-	
CREATE TABLE	EXCLUDE	
DROP INDEX	CONCURRENTLY	The subcommands on the left cannot be performed if a VCI is specified. This operation is not supported in VCI.
REINDEX	-	A VCI cannot be specified. This command is not required as VCI uses daemon's automatic maintenance to prevent disk space from increasing.

Client application command

- Operations that cannot be performed on relations containing a VCI

Command	Description
clusterdb	Clustering cannot be performed for tables that contain a VCI.
reindexdb	VCIs cannot be specified on the --index option.

10.2.2 Data Preload Feature

The first aggregation using VCI immediately after an instance is started may take time, because the VCI data has not been loaded to buffer.

Therefore, use the preload feature to load the VCI data to buffer in advance when performing VCI aggregation after an instance is started.

When using the preload feature, execute the function pgx_prewarm_vci to each VCI created with CREATE INDEX.

```
db01=# SELECT pgx_prewarm_vci('idx_vci');
```

Refer to "B.4 VCI Data Load Control Function" for information on pgx_prewarm_vci.

Chapter 11 Parallel Query

Fujitsu Enterprise Postgres enhances parallel queries, by taking into consideration the aspects below:

- CPU load calculation
- Increase of workers during runtime
- Statistics view displays the action state

11.1 CPU Load Calculation

There may be a case when the user tries to execute a parallel query but there is not enough CPU available.

Adding dynamic workers at this stage will provide no benefits - instead, it may add overhead due to context switching.

Fujitsu Enterprise Postgres takes into consideration the current CPU load when deciding on the number of workers for parallel query.

11.2 Increase of Workers during Runtime

This Fujitsu Enterprise Postgres enhancement allows systems to allocate additional workers during query execution (if there are workers available at the time). This improves query performance, which could otherwise starve of CPU if there were fewer or no workers when the query started.

Note

The ability to increase workers during runtime is available only with parallel query.

11.3 Statistics View Displays the Action State

You can check the action state by using the existing statistics collector. Monitoring the statistics enables you to check the action state of parallel processing through, for example, changes in the system load.

The following table describes the column added to the statistics views:

pg_stat_all_tables

Column	Data type	Description
parallel_query	bigint	Number of parallel query initialized in the table.

The parallel query column has also been added to the following views:

- pg_stat_sys_tables
- pg_stat_user_tables
- pg_stat_xact_all_tables
- pg_stat_xact_sys_tables
- pg_stat_xact_user_tables

See

Refer to "The Statistics Collector" in "Server Administration" in the PostgreSQL Documentation for information on the statistics collector and views.

Values will be set for the pgx_stat_sql view in Fujitsu Enterprise Postgres if parallel query is run.

See

Refer to "C.5 pgx_stat_sql" for information on pgx_stat_sql view.

Chapter 12 High-Speed Data Load

High-speed data load uses the pgx_loader command to load data from files at high speed into Fujitsu Enterprise Postgres.

Note

- This feature is available only in the Advanced Edition.
- This feature is not available in single-user mode. This is because in single-user mode instances run in a single process, and it cannot start parallel workers.

12.1 Installing High-Speed Data Load

This section describes how to install high-speed data load.

Installation flow

1. Deciding whether to Install
2. Estimating Resources
3. Setup

12.1.1 Deciding whether to Install

The feature achieves high speed data load by executing the COPY FROM command in parallel. If the database system is unable to use sufficient resources due to the feature using more resources than the COPY FROM command of PostgreSQL, load performance may be inferior to that of the COPY FROM command of PostgreSQL. Therefore, determine if the feature will be effective by considering the factors below before deciding to install.

Database server memory

If the value of shared_buffers in postgresql.conf is small, fewer data pages are cached to the shared memory of the database server. This will result in multiple parallel workers more often having to wait for write exclusive locks to the same data page. Moreover, the smaller the number of data pages, the more often the table expands. During table expansion, access to the table is exclusive (standby event name: extend), so write time increases. To cater for that, increase the value of shared_buffers.

See

The standby event name is stored in the wait_event column of the pg_stat_activity view. Refer to "wait_event Description" in "The Statistics Collector" in the PostgreSQL Documentation for details.

Frequency of checkpoints

If checkpoints are issued at short intervals, write performance is reduced. If the messages below are output to the server log during data writes, increase the values of max_wal_size and checkpoint_timeout in postgresql.conf to reduce the frequency of checkpoints.

Example

```
LOG:  checkpoints are occurring too frequently (19 seconds apart)
HINT:  Consider increasing the configuration parameter "max_wal_size".
```

12.1.2 Estimating Resources

Estimate the memory requirements for high-speed data load.

Up to 128 parallel workers to perform data load can be specified for this feature. The additional resources below are required depending on the number of parallel workers.

- Dynamic shared memory created during data load

  The feature creates shared memory and shared memory message queues during data load. These are used to send external data from the back end to the parallel workers, and for error notifications.

Note

If the value of shared_buffers in postgresql.conf is small, the system will often have to wait for write exclusive locks to the same data page (as described in "Database server memory" in "12.1.1 Deciding whether to Install"). Since input data cannot be loaded from the shared memory message queues during such waits, they will often be full. In these cases, it will not be possible to write to the shared memory message queues, resulting in degraded data load performance.

See

Refer to "High-Speed Data Load Memory Requirements" in the Installation and Setup Guide for Server for information on the formula for estimating memory requirements.

12.1.3 Setup

This section describes how to set up high-speed data load.

Setup flow

1. Setting Parameters
2. Installing the Extension

12.1.3.1 Setting Parameters

Set the parameters required for high-speed data load in postgresql.conf. After that, start or restart the instance.

The table below lists the postgresql.conf parameters that must be changed, and the values that must be added to their current values. After editing postgresql.conf, start or restart the instance.

Parameter	Setting	Required
max_prepared_transactions	Add the number of transactions that can be prepared by parallel workers during data load to the parameter's current value. The resulting value must be equal to or greater than the maximum number of parallel workers used with this feature.	Mandatory
max_worker_processes	Number of parallel workers to perform data load.	Mandatory
max_parallel_workers	Add the maximum number of parallel workers to be used in a parallel query by this feature to the parameter's current value. The resulting value must be equal to or greater than the number of parallel workers used with this feature.	Mandatory

Example

The example below shows how to configure 2 instances of high-speed data load being executed simultaneously using a degree of parallelism of 4.

```
max_prepared_transactions = 13  #Example if the initial value was 5: 5 + 2 x 4 = 13
max_worker_processes = 16        #Example if the initial value was 8: 8 + 2 x 4 = 16
max_parallel_workers = 12        #Example if the initial value was 4: 4 + 2 x 4 = 12
```

Note

As shown in the example above, set the value of max_prepared_transactions, max_worker_processes and max_parallel_workers multiplied by the number of instances of this feature executed simultaneously.

The table below lists the postgresql.conf parameters that must also be checked.

Parameter	Setting	Required
dynamic_shared_memory_type	Implementation of dynamic shared memory to be used by the instance. The default value is recommended.	Mandatory

See

Refer to "Resource Consumption" in the PostgreSQL Documentation for information on the parameters.

12.1.3.2 Installing the Extension

Execute CREATE EXTENSION to install the high-speed data load extension. The extension needs to be installed on each database.

Example

The example below installs the extension on the "postgres" database.

```
postgres=# CREATE EXTENSION pgx_loader;
CREATE EXTENSION
```

Note

- Only superusers can install the high-speed data load extension.
- The high-speed data load extension can only be installed on the public schema.

12.2 Using High-Speed Data Load

This section describes how to use high-speed data load.

12.2.1 Loading Data

To load data from a file into a Fujitsu Enterprise Postgres table, execute the pgx_loader command in load mode.

Example

The example below loads the file /path/to/data.csv (2000 records) into table tbl using a degree of parallelism of 3.

```
$ pgx_loader load -j 3 -c "COPY tbl FROM '/path/to/data.csv' WITH CSV"
LOAD 2000
```

Point

If an external file contains data that violates the format or constraints, the data load may fail partway through, resulting in delays for routine tasks such as nightly batch processing. Therefore, it is recommended to remove the invalid data before executing the data load.

Note

The data inserted using this feature is dumped as a COPY command by the pg_dump command and the pg_dumpall command.

See

- Refer to "pgx_loader" in the Reference for information on the command.
- Refer to "COPY" in the PostgreSQL Documentation for information on the deployment destination and access privileges for external files.

12.2.2 Checking Progress

If you are performing a data load with a large external file as input, you can verify that the process is continuing by getting progress information during the load. Progress information can be obtained from the pgx_stat_progress_loader view. This view displays the sum of the progress information of the back-end process and the number of parallel worker processes. Search the pgx_stat_progress_loader view, for example, with a SELECT statement, to locate the appropriate row. After running the pgx_loader command, look in the pg_stat_activity view and locate a row in the pgx_stat_progress_loader view with the PID obtained.

Example

1. See the pg_stat_activity view. (9311 for back-end processes, 9312, 9313, 9314 for worker processes)

```
postgres=# select pid, application_name, backend_type from pg_stat_activity
pid  | application_name |         backend_type
-----+------------------+------------------------------
6216 |                  | autovacuum launcher
6218 |                  | logical replication launcher
6271 | psql             | client backend
9311 | pgx_loader       | client backend
9312 |                  | parallel loader for PID 9311
9313 |                  | parallel loader for PID 9311
9314 |                  | parallel loader for PID 9311
6214 |                  | background writer
6213 |                  | checkpointer
6215 |                  | walwriter
```

2. Check the information in the pgx_stat_progress_loader view.

```
postgres=# SELECT * FROM pgx_stat_progress_loader
pid  | datid | datname | relid | command  | type | bytes_processed | bytes_total |
tuples_processed | tuples_excluded
-----+-------+---------+-------+----------+------+-----------------+-------------
+------------------+----------------
9311 |  222  | testdb  | 333   |COPY FROM | FILE |        192000   |    450000   |        189000
|           3000
```

Refer to "C.7 pgx_stat_progress_loader" for information on the pgx_stat_progress_loader view.

Note

When you run the pgx_loader command, the PostgreSQL pg_stat_progress_copy view prints the progress of the back-end process and the number of parallel worker processes on each line. The backend process progress information tuples_processed, tuples_excluded is 0. Also, bytes_processed and bytes_total for worker processes are 0.

```
postgres=# SELECT * FROM pg_stat_progress_copy
pid  | datid | datname | relid | command  | type | bytes_processed | bytes_total | tuples_processed |
tuples_excluded
-----+-------+---------+-------+----------+------+-----------------+-------------+------------------
+---------------
9311 |  222  | testdb  | 333   |COPY FROM | FILE |         192000  |    450000   |              0
|               0
9312 |  222  | testdb  | 333   |COPY FROM | FILE |              0  |         0   |          63000
|            1000
9313 |  222  | testdb  | 333   |COPY FROM | FILE |              0  |         0   |          63000
|            1000
9314 |  222  | testdb  | 333   |COPY FROM | FILE |              0  |         0   |          63000
|            1000
```

Refer to "pg_stat_progress_copy View" in the PostgreSQL Documentation for information on the pg_stat_progress_copy view.

12.2.3 Recovering from a Data Load that Ended Abnormally

If a system interruption such as a server failure occurs while high-speed data load is being performed, transactions prepared using this feature may be changed to the in-doubt state. At that point, resources occupied by the transaction will be locked, and access to the relevant resources from other transactions will be blocked, rendering them unusable.

In such cases, check transactions that are in an in-doubt state, and resolve them.

Checking for in-doubt transactions

This section describes how to check for in-doubt transactions.

1. Refer to the pgx_loader_state table in the pgx_loader schema.

   Retrieve the global transaction identifier (gid column) of in-doubt transactions. In-doubt transactions will contain "rollback" in the column "state".

   #### Example

   The example below retrieves the global transaction identifier (gid) of in-doubt transactions performed by the database role myrole and that used table tbl. The retrieved global transaction identifiers pgx_loader:9589 and pgx_loader:9590 identify in-doubt transactions.

   ```
   postgres=# SELECT gid, state FROM pgx_loader.pgx_loader_state WHERE
   postgres-# role_oid IN (SELECT oid FROM pg_roles WHERE rolname = 'myrole') AND
   postgres-# relation_oid IN (SELECT relid FROM pg_stat_all_tables WHERE
   postgres-# relname = 'tbl');
          gid       | state
   -----------------+-------------
    pgx_loader:9590 | rollback
    pgx_loader:9591 | commit
    pgx_loader:9589 | rollback
   (3 rows)
   ```

2. Refer to the pg_prepared_xacts system view.

   Check if the in-doubt transactions retrieved above exist.

Example

The example below checks if in-doubt transactions with the global transaction identifiers pgx_loader:9589 and pgx_loader:9590 exist.

```
postgres=# SELECT gid FROM pg_prepared_xacts WHERE gid IN ('pgx_loader:9589','pgx_loader:9590');
       gid
------------------
 pgx_loader:9590
 pgx_loader:9589
(2 rows)
```

See

Refer to "F.1 pgx_loader_state" for information on the pgx_loader_state table.

Resolving in-doubt transactions

Execute the pgx_loader command in recovery mode to resolve in-doubt transactions.

After executing the pgx_loader command in recovery mode, perform the procedure described in "Checking for in-doubt transactions" to check if the in-doubt transactions have been resolved.

Example

The example below completes the in-doubt transactions prepared for table tbl.

```
$ pgx_loader recovery -t tbl
```

Point

The recovery mode of the pgx_loader command only resolves transactions prepared by high-speed data load. For transactions prepared by an application using distributed transactions other than this feature, follow the procedure described in "16.13 Actions in Response to Error in a Distributed Transaction".

12.3 Removing High-Speed Data Load

This section describes how to remove high-speed data load.

12.3.1 Removing the Extension

Execute DROP EXTENSION to remove the high-speed data load extension. The extension needs to be removed on each database.

Example

The example below removes the extension on the "postgres" database.

```
postgres=# DROP EXTENSION pgx_loader;
DROP EXTENSION
```

Note

- The information required for operation of high-speed data load is stored in the pgx_loader_state table of the pgx_loader schema. Do not remove the high-speed data load extension if the pgx_loader_state table is not empty.
- Only superusers can remove the high-speed data load extension.
- The high-speed data load extension can only be removed on the public schema.

Chapter 13 Global Meta Cache

The Global Meta Cache (GMC) feature loads a meta cache into shared memory using the pgx_global_metacache parameter. This reduces the amount of memory required throughout the system.

Note

This feature can only be used in Advanced Edition.

13.1 Usage

Describes how to use the Global Meta Cache feature.

13.1.1 Deciding Whether to Enable the Global Meta Cache Feature

Global Meta Cache is a mechanism for sharing meta caches between processes, so it works well on systems with a high number of resources accessed and SQL connections. The number of resources is primarily the number of tables accessed by a process, the number of indexes, or the total number of all columns in all tables accessed.

In particular, consider using Global Meta Cache if the total size of the meta cache for each process exceeds the amount of installed memory, or takes up a large portion of that memory, thereby squeezing memory allocations to the database cache or the Operating system file cache. Using Global Meta Cache may increase the time it takes to execute a single SQL to reference a meta cache on shared memory, but you can expect a greater benefit from being able to allocate more memory, such as for the database cache.

If performance degradation using Global Meta Cache is not acceptable, you may want to limit the number of tables accessed by a process.

13.1.2 Estimating Memory for Global Meta Cache

To enable the Global Meta Cache feature, the pgx_global_metacache parameter must specify an upper limit on the size of the shared memory (Hereinafter, the GMC area) dedicated to Global Meta Cahche. Ideally, this upper limit should be the size estimated in “Appendix A Parameters”. Values lower than this can still work, but refer to "13.1.3 How the GMC Memory Area Is Used" on using the GMC area to understand the disadvantages.

13.1.3 How the GMC Memory Area Is Used

At startup, the memory for the GMC area is not used much, but the GMC area grows as new meta caches are placed in the GMC area. If it does, it discards any meta caches that the system determines are not heavily used and places a new one in the GMC area.

Therefore, the GMC area will work even if it is smaller than the estimate, but the meta cache will be regenerated if the discarded meta cache needs to be reused. Note that if this happens frequently, it will degrade overall performance.

With this in mind, it may not be a problem if, for example, the tables to be accessed are different depending on the time zone, and the degradation of the time zone immediately after the change is acceptable.

In any case, be sure to test and tune the system thoroughly before running it.

13.1.4 Enabling the Global Meta Cache Feature

To Enable the Global Meta Cache feature edit the postgresql.conf file and set the pgx_global_metacahe parameter. Restarting the instance after editing the postgresql.conf file is required. Refer to "Appendix A Parameters" for information on the parameters.

Parameter Name	Description
pgx_global_metacache	Specify the maximum amount of memory for the GMC area on shared memory. When it’s set to 0 (default value), the Global Meta Cache feature is disabled. When enabled, the minimum value allowed is 10MB.

When the cache is created, if the total amount of meta caches on shared memory exceeds the value specified by pgx_global_metacache, the inactive, unreferenced meta caches are removed from the GMC area. Note that if all GMC are in use and the cache cannot be created in the GMC area, the cache is temporarily created in the local memory of the backend process.

Example

Here is an example postgresql.conf configuration:

```
pgx_global_metacache = 800 MB
```

Wait Events

The Global Meta Cache feature may cause wait events. Wait events are identified in the wait_event column of the pg_stat_activity view. GMC specific wait events are described below.

[GMC Feature Wait Events]

Wait Event Type	Wait Event Name	Description
LWLock	GlobalCatcache	Waiting to find, add, and remove meta caches in the GMC area.
IPC	GMCSweep	Waiting to select a meta cache that can be deleted when GMC space is low. If the GMC is fully referencing and there is no deletable meta cache, it is waiting for the reference to be removed and a deletable meta cache to be selected.

Note

If GMCSweep is happened frequently, increase the pgx_global_metacache setting.

See

Refer to "Viewing Statistics" in the PostgreSQL Documentation for information on the pg_stat_activity view.

13.1.5 Estimating Resources

Refer to "Global Meta Cache Memory Requirements" in the Installation and Setup Guide for Server for formulas to estimate the amount of memory used by the Global Meta Cache feature.

13.2 Statistics

Describes the statistics for the Global Meta Cache feature.

13.2.1 System View

You can check the cache hit ratio and size of the GMC area in the system view pgx_stat_gmc. Refer to "C.6 pgx_stat_gmc" for information on the columns.

If the cache hit ratio is low and the current memory usage is close to pgx_global_metacache, increase the pgx_global_metacache setting because performance may be degraded.

Refer to "8.6 Monitoring Database Activity" in the Operations Guide for information on the statistics.

Chapter 14 Local Meta Cache Limit

Local Meta Cache Limit feature limits the size of a Local Meta Cache by removing it if it has not been accessed for a long time.

Note

This feature is available only in the Advanced Edition.

14.1 Usage

Describes how to use the Local Meta Cache Limit feature.

14.1.1 Deciding Whether to Enable the Local Meta Cache Limit Feature

Refer to "Appendix A Parameters", after estimating the total amount of memory to be used as the catalog cache and relation cache, when the total amount of memory exceeds the amount of installed memory or occupies a large amount of installed memory, consider using this feature.

This feature adds the action of discarding the meta cache that has been held permanently. If you attempt to refer to a destroyed meta cache again, the meta cache is recreated, so using this feature will result in poor performance compared to not using it.

Therefore, read the following to understand how to discard a meta cache.

- 14.1.3 Cache Removal when Local Meta Cache Limit is Enabled
- 14.1.4 Performance Impact and Parameter Tuning of the Local Meta Cache Limit Feature
- Parameters for the Local Meta Cache Limit feature

How to set the upper limit with these considerations is described in detail in the estimation formula in "Appendix A Parameters".

14.1.2 How to Set Parameters for the Local Meta Cache Limit Feature

To enable the Local Meta Cache Limit feature, set the pgx_catalog_cache_max_size and pgx_relation_cache_max_size parameters.

Parameter Name	Description
pgx_catalog_cache_max_size	Specify the maximum amount of memory that the backend process should use as the catalog cache. You can enable catalog cache removal by setting it to 8 KB or more. When it is set to 0 (default value), the catalog cache removal is disabled.
pgx_relation_cache_max_size	Specify the maximum amount of memory that the backend process should use as the relation cache. You can enable relation cache removal by setting it to 8 KB or more. When it is set to 0 (default value), the relation cache removal is disabled.

Example

Here is an example postgresql.conf configuration:

```
pgx_catalog_cache_max_size = 1MB
pgx_relation_cache_max_size = 1MB
```

14.1.3 Cache Removal when Local Meta Cache Limit is Enabled

When this feature is enabled, the caching strategy is to keep the cache as long as possible within the specified upper limit. If holding a new cache exceeds the limit, consider locality of reference and remove the cache from the one with the longest unreferenced time.

However, because the cache used by active transactions cannot be removed, if a transaction uses a large number of caches, the cache may be held above the limit. In this case, remove the all caches at the end of the transaction. This is necessary to free up memory.

In PostgreSQL, in order to acquire memory at high speed, a memory block of a certain size is acquired from the OS, and a small memory is cut out from the block and used. The memory for the metacache is cut out in the same way. Therefore, it is possible to return the memory block to the OS by destroying all the meta caches scattered throughout the memory block. When this happens, the next SQL execution will be slowed down due to the re-creation of the meta cache. Therefore, upper limit of feature should be set to a value larger than the size of the meta cache used by at least one transaction.

When the size of the meta cache exceeds the upper limit, the following message is output:

```
WARNING: could not reduce Cat/RelCacheMemoryContext size to AA kilobytes, reduced to BB kilobytes
HINT: consider increasing the configuration parameter pgx_catalog/relation_cache_max_size
```

(***AA***: Upper limit, ***BB***: Amount of memory actually used)

CatCacheMemoryContext and RelCacheMemoryContext are memory areas for storing the catalog cache and relation cache, respectively. If this message is output, consider increasing the upper limit.

If the memory consumption by the backend process exceeds the allowable value by increasing the upper limit, reconsider the SQL to be executed, such as reducing the number of tables accessed in one transaction, or add memory adjust to the amount of memory used.

14.1.4 Performance Impact and Parameter Tuning of the Local Meta Cache Limit Feature

By observing how much meta cache regeneration is taking place, you can determine if the low upper limit is the cause of the failure to achieve the desired performance.

From the message below, calculate the cache hit ratio as follows:

```
Cache hit ratio = Number of cache hits ÷ Number of times the cache was searched
```

If the cache hit ratio is 80% or higher, this feature will not be the main factor that impedes performance. If not, raise the upper limit and see if performance can reach the goal. In doing so, first try to shift the focus of allocation to the relations cache. This is because when executing SQL, the relation cache generated based on the catalog cache is mainly referenced, so it is advantageous to leave a large amount of relation cache.

```
Catalog cache:catalog cache hit stats: search XX, hits YY
Relation cache:relation cache hit stats: search XX, hits YY
```

(***XX***: Number of times the cache was searched, ***YY***: Number of cache hits)

This message is printed when the transaction ends. However, if you output the message frequently, the performance will be degraded by itself, so you can adjust the output interval with the following parameters.

Parameter Name	Description
pgx_cache_hit_log_interval	When the transaction ends, if the time set in this parameter has elapsed since the previous message was output, the message is output. If set to 0, a message will be output each time the transaction ends. Setting -1 disables the output. The default value is 10min. Even if pgx_catalog_cache_max_size and pgx_relation_cache_max_size are disabled, the message output of the corresponding cache will be invalid. Immediately after connecting to the server, a small transaction occurs before the request from the user application, such as for user authentication. Since it is meaningless to know the hit ratio for these, a message is output at the end of the transaction that started after the time set in this parameter has elapsed after connecting to the server. For the same reason, setting a small value such as 0 may result in a message being printed at the end of such a small transaction. You can check which transaction the message corresponds to from the information output at the beginning. This information depends on the setting of the parameter log_line_prefix.

Example

Here is an example postgresql.conf configuration:

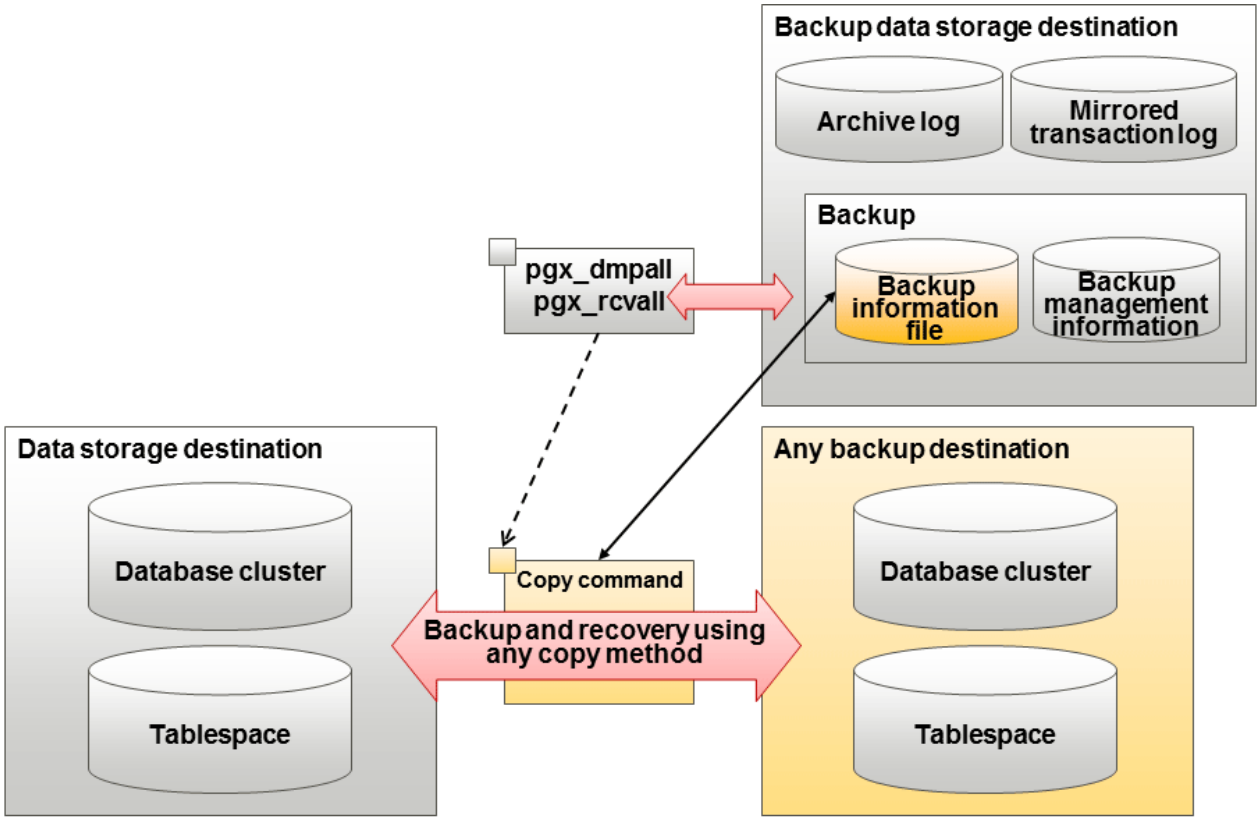
```
pgx_cache_hit_log_interval= 30min
```

Chapter 15 Backup/Recovery Using the Copy Command

By using a copy command created by the user, the pgx_dmpall command and the pgx_rcvall command can perform backup to any destination and can perform recovery from any destination using any copy method.

Copy commands must be created in advance as executable scripts for the user to implement the copy process on database clusters and tablespaces, and are called when executing the pgx_dmpall and pgx_rcvall commands.

This appendix describes backup/recovery using the copy command.



Point

- By using the high-speed copy feature of the storage device to copy the data storage destination, the processing time for backup of large databases can be greatly reduced.
- It is also possible to back up only some tablespaces using the copy command. However, database resources not backed up using the copy command are still backed up to the backup data storage destination.

Note

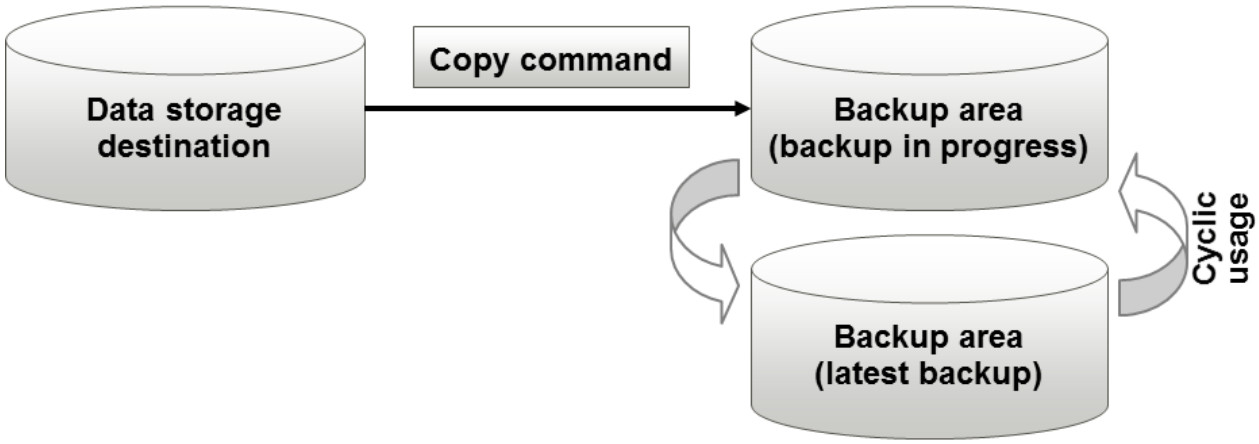
Both the backup data storage destination and the optional backup destination are necessary for recovery - if they are located in secondary media, combined management of these is necessary.

15.1 Configuration of the Copy Command

This section describes the configuration of the copy command for backup and recovery.

Cyclic usage of the backup area

Prepare two backup areas for the copy command in case an issue affects the data storage destination during backup. The copy command performs backup while cyclically using these backup areas.

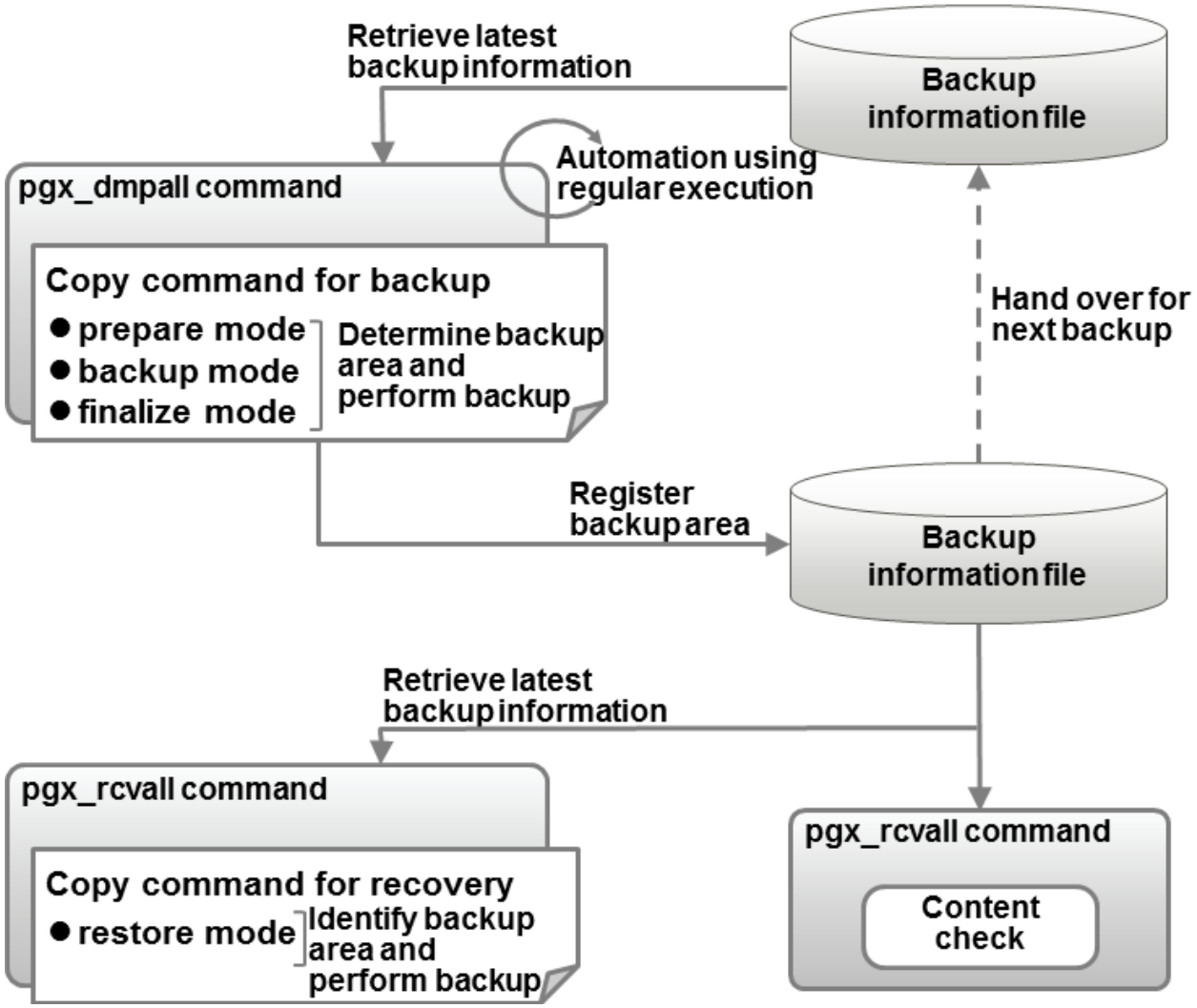


Note

The backup data storage destination cannot be used as these backup areas used by the copy command.

Backup using the backup information file

The copy command must determine the backup destination on each backup, as it is necessary to cycle through the backup areas. Backup can be automated by using the backup information file, which contains information about the backup destination.



Information

The backup information file is prepared in the backup data storage destination by the pgx_dmpall command, and contains information that can be read or updated by the copy command. This file is managed by associating it with the latest backup successfully completed by the pgx_dmpall command, so the latest backup information relating to the copy command registered by the user can be retrieved. Additionally, the content of the backup information file can be displayed using the pgx_rcvall command.

Configuration of the copy command for backup

The pgx_dmpall command calls the copy command for backup after execution for the three modes below. It is therefore necessary for the copy command for backup to implement the required processing for each of the modes.

- prepare mode

  Determines which of the two backup areas will be used for the current backup.

  The backup area to be used for the current backup is determined by reading the information relating to the latest backup destination where the backup information file was written to during the previous backup.

- backup mode

  Performs backup on the backup area determined by prepare mode, using any copy method.

- finalize mode

  Writes information relating to the destination of the current backup to the backup information file.

This enables the prepare mode to check the destination of the previous backup during the next backup.

Note

The user can use any method to hand over backup information between modes within the copy command, such as creating temporary files.

Configuration of the copy command for recovery

The pgx_rcvall command calls the copy command for recovery for the mode below. It is therefore necessary for the copy command for recovery to implement the required processing for the mode.

- restore mode

  Any copy method can be used to implement restore from the backup destination retrieved using the copy command for backup.

Point

By referring to the mode assigned to the copy command as an argument, backup and recovery can be implemented using a single copy command.

Example

Using a bash script

```
case $1 in
    prepare)
        processingRequiredForPrepareMode
        ;;
    backup)
        processingRequiredForBackupMode
        ;;
    finalize)
         processingRequiredForFinalizeMode
        ;;
    restore)
         processingRequiredForRestoreMode
        ;;
esac
```

Point

- A sample script that backs up the database cluster and tablespace directory to a specific directory is supplied to demonstrate how to write a copy command.

  The sample is stored in the directory below:

  ```
  /installDir/share/copy_command.archive.sh.sample
  ```

- A sample script that uses OPC (an advanced copy feature of Fujitsu Storage ETERNUS disk array) is supplied. Refer to "Appendix J Copy Command Samples that Use the Advanced Copy Feature of the ETERNUS Disk Array" for details.

  The samples cannot be used on SLES 12 and SLES 15.

15.2 Backup Using the Copy Command

To perform backup using the copy command, in addition to performing the standard backup procedure, it is also necessary to create a copy command, and then execute the pgx_dmpall command specifying it. This section describes the procedure specific to using the copy command.

Preparing for backup

You must prepare for backup before actually starting the backup process.

Perform the following procedure:

1. Determine the database resources to be backed up

   Determine the database resources to be backed up using the copy command. The copy command can back up the following resources:

   - Database cluster
   - Tablespace

   To back up only some tablespaces, create a file listing them. This file is not necessary to back up all tablespaces.

   Example

   To back up only tablespaces tblspc1 and tblspc2

   ```
   tblspc1
   tblspc2
   ```

2. Prepare a backup area

   Prepare a backup area to save the database resources to be backed up, as determined in step 1.

3. Create the copy command

   Create the copy commands for backup and recovery. Refer to "15.4 Copy Command Interface" for details.

Performing backup

Execute the pgx_dmpall command with the -Y option specifying the full path of the copy command for backup created in step 3 of preparation for backup.

The example below backs up only some tablespaces, but not the database cluster, using the copy command.

Example

```
$ pgx_dmpall -D /database/inst1 -Y '/database/command/backup.sh'
  --exclude-copy-cluster -P '/database/command/tablespace_list.txt'
```

Point

- To exclude up the database cluster from backup using the copy command, specify the --exclude-copy-cluster option.
- To back up only some tablespaces using the copy command, use the -P option specifying the full path of the file created in step 1 of preparation for backup.

See

- Refer to "pgx_dmpall" in the Reference for information on the command.

Checking backup status

Use the pgx_rcvall command to check the backup status.

Execute the pgx_rcvall command with the -l option specified to output backup data information. If backup was performed using the copy command, the resources backed up using the copy command will also be output.

Example

```
$ pgx_rcvall -l -D /database/inst1
Date                Status   Dir                                  Resources backed up by the copy command
2022-03-01 13:30:40 COMPLETE /backup/inst1/2022-03-01_13-30-40 pg_data,dbspace,indexspace
```

15.3 Recovery Using the Copy Command

To perform recovery using the copy command, in addition to performing the standard recovery procedure, it is also necessary to create a copy command, and then execute the pgx_rcvall command specifying it. This section describes the procedure specific to using the copy command.

Determining the backup area of the latest backup

Check the backup information file to determine the backup area used for the latest backup, and confirm that it is in a recoverable state.

Execute the pgx_rcvall command with the --view-results-of-copying option to output the content of the backup information file.

Example

```
$ pgx_rcvall -D /database/inst1 --view-results-of-copying
```

Perform recovery

Execute the pgx_rcvall command with the -Y option specifying the full path of the copy command for recovery created in step 3 of the preparation for backup described in "15.2 Backup Using the Copy Command".

The example below recover only some tablespaces, but not the database cluster, using the copy command.

Example

```
$ pgx_rcvall -D /database/inst1 -B /backup/inst1 -Y '/database/command/recovery.sh'
```

Point

If the latest backup was performed using the copy command, the pgx_rcvall command automatically recognizes which database resources were backed up using the copy command, or whether resources were backed up to the backup data storage destination. Therefore, recovery can be performed by simply executing the pgx_rcvall command specifying the copy command for recovery.

See

Refer to "pgx_rcvall" in the Reference for information on the command.

15.4 Copy Command Interface

The following types of copy command are available:

- Copy command for backup
- Copy command for recovery

This appendix describes the interface of each copy command.

15.4.1 Copy Command for Backup

Feature

User exit (for the copy command) called from the pgx_dmpall command.

Format

The syntax for calling the copy command from the pgx_dmpall command is described below.

If the operation mode is "prepare"

```
copyCommandName prepare 'pathOfBackupInfoFile' 'pathOfBackupTargetListFile'
```

If the operation mode is "backup"

```
copyCommandName backup
```

If the operation mode is "finalize"

```
copyCommandName finalize 'pathOfBackupInfoFile'
```

Argument

- Operation mode

Mode	Description
prepare	Implements the preparation process for backing up using the copy command. Called before the PostgreSQL online backup mode is started.
backup	Implements the backup process. Called during the PostgreSQL online backup mode.
finalize	Implements the backup completion process. Called after the PostgreSQL online backup mode is completed.

Point

If using high-speed copy of the storage device (which performs high-speed retrieval of snapshots and copies data to different physical areas), it is possible to invoke the snapshot process in backup mode, and the copy process in finalize mode.

- Full path of the backup information file

  Full path of the backup information file of the latest backup, enclosed in single quotation marks. If a backup has not been performed, specify '-'.

- Full path of the backup target list file

  Full path of the file containing the resources to be backed up using the copy command, enclosed in single quotation marks. One of the following is described in each resource name.

Resource	Description
Database cluster	pg_data
Tablespace	Tablespace name

Example

To back up the database cluster and the tablespaces dbspace and indexspace using the copy command, the file should contain the following:

```
pg_data
dbspace
indexspace
```

Information

The encoding of resource names output to the backup target list file by the pgx_dmpall command is the encoding used when this command connects to the database with auto specified for the client_encoding parameter, and is dependent on the locale at the time of command execution.

The number of arguments vary depending on operation mode. The argument of each operation mode is as follows.

Operation mode	First argument	Second argument	Third argument
prepare	Operation mode	Backup information file path name	Backup target list file path name
backup		None	None
finalize		Backup information file path name	

Additionally, the access permissions for the backup information file and backup target list file are different depending on the operation mode. The access permissions of each operation mode are as follows.

Operation mode	Backup information file	Backup target list file
prepare	Can be viewed by the instance administrator only	Can be viewed by the instance administrator only
backup	-	-
finalize	Can be viewed and updated by the instance administrator only	-

Return value

Return value	Description
0	Normal end The pgx_dmpall command continues processing.
Other than 0	Abnormal end The pgx_dmpall command terminates in error.

Description

- The copy command operates with the privileges of the operating system user who executed the pgx_dmpall command. Therefore, grant copy command execution privileges to users who will execute the pgx_dmpall command. Additionally, have the copy command change users as necessary.

- To write to the backup information file, use a method such as redirection from the copy command.
- Because the copy command is called for each mode, implement all processing for each one.
- To copy multiple resources simultaneously, have the copy command copy them in parallel.

Note

- The backup information file and backup target list file cannot be deleted. Additionally, the privileges cannot be changed.
- Standard output and standard error output of the copy command are output to the terminal where the pgx_dmpall command was executed.
- If the copy command becomes unresponsive, the pgx_dmpall command will also become unresponsive. If the copy command is deemed to be unresponsive by the operating system, use an operating system command to forcibly stop it.
- Output the copy command execution trace and the result to a temporary file, so that if it terminates in error, the cause can be investigated at a later time.
- For prepare mode only, it is possible to use the PostgreSQL client application to access the database using the copy command. For all other modes, do not execute Fujitsu Enterprise Postgres commands or PostgreSQL applications.
- Enable the fsync parameter in postgresql.conf, because data on the shared memory buffer needs to have been already written to disk when backup starts.

15.4.2 Copy Command for Recovery

Feature

User exit (for the copy command) called from the pgx_rcvall command.

Format

The syntax for calling the copy command from the pgx_rcvall command is described below.

```
copyCommandName restore 'pathOfBackupInfoFile' 'pathOfBackupTargetListFile'
```

Argument

- Operation mode

Mode	Description
restore	Performs restore.

- Full path of the backup information file

  Full path of the backup information file, enclosed in single quotation marks.

- Full path of the backup target list file

  Full path of the file containing the resources to be restored using the copy command, enclosed in single quotation marks.

The access permissions for the backup information file and backup target list file are as below.

Backup information file	Backup target list file
Can be viewed by the instance administrator only	Can be viewed by the instance administrator only

Return value

Return value	Description
0	Normal end The pgx_rcvall command continues processing.
Other than 0	Abnormal end The pgx_rcvall command terminates in error.

Description

- The copy command operates with the privileges of the operating system user who executed the pgx_rcvall command. Therefore, grant copy command execution privileges to users who will execute the pgx_rcvall command. Additionally, have the copy command change users as necessary.
- The copy command is called once only in restore mode.
- To copy multiple resources simultaneously, have the copy command copy them in parallel.

Note

- The backup information file and backup target list file cannot be deleted. Additionally, the privileges cannot be changed.
- Standard output and standard error output of the copy command are output to the terminal where the pgx_rcvall command was executed.
- If the copy command becomes unresponsive, the pgx_rcvall command will also become unresponsive. If the status of the copy command is deemed to be unresponsive by the operating system, use an operating system command to forcibly stop it.
- Output the copy command execution trace and the result to a temporary file, so that if it terminates in error, the cause can be investigated at a later time.
- Do not execute Fujitsu Enterprise Postgres commands or PostgreSQL applications in the copy command.
- There may be files and directories not required for recovery using the archive log included in the backup, such as postmaster.pid, pg_wal/*subdirectory* and pg_replslot in the database cluster. If such unnecessary files and directories exist, have the copy command delete them after the restore.

Chapter 16 Actions when an Error Occurs

This chapter describes the actions to take when an error occurs in the database or an application, while Fujitsu Enterprise Postgres is operating.

Depending on the type of error, it may be necessary to recover the database cluster. The recovery process recovers the following resources:

- Data storage destination
- Transaction log storage destination (if the transaction log is stored in a separate disk from the data storage destination)
- Backup data storage destination

Note

Even if a disk is not defective, the same input-output error messages, as those generated when the disk is defective, may be output. The recovery actions differ for these error messages.

Check the status of the disk, and select one of the following actions:

- If the disk is defective

  Refer to "16.1 Recovering from Disk Failure (Hardware)", and take actions accordingly.

- If the disk is not defective

  Refer to "16.14 I/O Errors Other than Disk Failure", and take actions accordingly.

  A few examples of errors generated even if the disk is not defective include:

  - Network error with an external disk
  - Errors caused by power failure or mounting issues

Determining the cause of an error

If an error occurs, refer to the WebAdmin message and the server log, and determine the cause of the error.

See

Refer to "Configuring Parameters" in the Installation and Setup Guide for Server for information on server logs.

Approximate recovery time

The formulas for deriving the approximate recovery time of resources in each directory are given below.

If using the copy command with the pgx_rcvall command, the recovery time will depend on the implementation of the copy command.

- Data storage destination or transaction log storage destination

```
Recovery time = (usageByTheDataStorageDestination + usageByTheTransactionLogStorageDestination) /
diskWritePerformance x 1.5
```

  - *usageByTheDataStorageDestination*: Disk space used by the database cluster
  - *usageByTheTransactionLogStorageDestination*: Disk space used by the transaction log stored outside the database cluster
  - *diskWritePerformance*: Measured maximum data volume (bytes/second) that can be written per second in the system environment where the operation is performed
  - 1.5: Coefficient assuming the time excluding disk write, which is the most time-consuming step

- Backup data storage destination

```
Recovery time = usageByTheBackupDataStorageDestination / diskWritePerformance x 1.5
```

- *usageByTheBackupDataStorageDestination*: Disk space used by the backup data
- *diskWritePerformance*: Measured maximum data volume (bytes/second) that can be written per second in the system environment where the operation is performed
- 1.5: Coefficient assuming the time excluding disk write, which is the most time-consuming step

16.1 Recovering from Disk Failure (Hardware)

This section describes how to recover database clusters to a point immediately before failure, if a hardware failure occurs in the data storage disk or the backup data storage disk.

There are two methods of recovery:

- 16.1.1 Using WebAdmin
- 16.1.2 Using Server Command

Point

Back up the database cluster after recovering it. Backup deletes obsolete archive logs (transaction logs copied to the backup data storage destination), freeing up disk space and reducing the recovery time.

16.1.1 Using WebAdmin

Recover the database cluster by following the appropriate recovery procedure below for the disk where the failure occurred.

Note

Recovery operation cannot be performed on an instance that is part of a streaming replication cluster in standby mode.

If disk failure occurs on a standby instance, it may be necessary to delete and re-create the instance.

Recovery operation can be performed on an instance that is part of a streaming replication cluster in "Master" mode. If a recovery operation is performed on a master instance, it will break the replication cluster and streaming replication will stop between the master instance and all its standby instances. In such an event, the standby instances can be promoted to standalone instances or can be deleted and re-created.

If failure occurred in the data storage disk or the transaction log storage disk

Follow the procedure below to recover the data storage disk or the transaction log storage disk.

1. Stop applications

   Stop applications that are using the database.

2. Stop the instance

   Stop the instance. Refer to "2.1.1 Using WebAdmin" for information on how to stop an instance. WebAdmin automatically stops instances if recovery of the database cluster is performed without stopping the instance.

3. Recover the failed disk

   Replace the disk, and then recover the volume configuration information.

4. Create a tablespace directory

   If a tablespace was defined after backup, create a directory for it.

5. Recover the keystore, and enable automatic opening of the keystore

   Do the following if the data in the database has been encrypted:

   - Restore the keystore to its state at the time of the database backup.
   - Enable automatic opening of the keystore.

6. Recover the database cluster

   Log in to WebAdmin, and in the [Instances] tab, click [Solution] for the error message in the lower-right corner.

7. Run recovery

   In the [Restore Instance] dialog box, click [Yes].

   Instance restore is performed. An instance is automatically started when recovery is successful.

8. Resume applications

   Resume applications that are using the database.

Point

WebAdmin may be unable to detect disk errors, depending on how the error occurred.

If this happens, refer to "16.10.3 Other Errors" to perform recovery.

If failure occurred on the backup data storage disk

Follow the procedure below to recover the backup data storage disk.

1. Recover the failed disk

   Replace the disk, and then recover the volume configuration information.

2. Recover the backup data

   Log in to WebAdmin, and in the [Instances] tab, click [Solution] for the error message.

3. Run backup

   Perform backup to enable recovery of the backup data. In the [Backup] dialog box, click [Yes]. The backup is performed. An instance is automatically started when backup is performed.

Point

If you click [Recheck the status], the resources in the data storage destination and the backup data storage destination are reconfirmed. As a result, the following occurs:

- If an error is not detected

  The status of the data storage destination and the backup data storage destination returns to normal, and it is possible to perform operations as usual.

- If an error is detected

  An error message is displayed in the message list again. Click [Solution], and resolve the problem by following the resolution for the cause of the error displayed in the dialog box.

16.1.2 Using Server Command

Recover the database cluster by following the appropriate recovery procedure below for the disk where the failure occurred.

If failure occurred on the data storage disk or the transaction log storage directory

Follow the procedure below to recover the data storage disk or the transaction log storage directory.

1. Stop applications

   Stop applications that are using the database.

2. Stop the instance

   Stop the instance, refer to "2.1.2 Using Server Commands" for details.

If the instance fails to stop, refer to "16.11 Actions in Response to Failure to Stop an Instance".

3. Recover the failed disk

   Replace the disk, and then recover the volume configuration information.

4. Create a storage destination directory

   - If failure occurred on the data storage disk
     Create a data storage destination directory. If a tablespace was defined, also create a directory for it.
   - If failure occurred on the translation log storage disk
     Create a transaction log storage destination directory.

   Example

   To create a data storage destination directory:

   ```
   $ mkdir /database/inst1
   $ chown fsepuser:fsepuser /database/inst1
   $ chmod 700 /database/inst1
   ```

   See

   Refer to "Preparing Directories to Deploy Resources" under "Setup" in the Installation and Setup Guide for Server for information on how to create a storage directory.

5. Recover the keystore, and enable automatic opening of the keystore

   When the data in the database has been encrypted, restore the keystore to its state at the time of the database backup. Configure automatic opening of the keystore as necessary.

6. Recover the database cluster

   Recover the database cluster using the backup data.

   Specify the following in the pgx_rcvall command:

   - Specify the data storage location in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
   - Specify the backup data storage location in the -B option.

   Example

   ```
   > pgx_rcvall -D /database/inst1 -B /backup/inst1
   ```

   Note

   If recovery fails, remove the cause of the error in accordance with the displayed error message and then re-execute the pgx_rcvall command.

   If the message "pgx_rcvall: an error occurred during recovery" is displayed, then the log recorded when recovery was executed is output after this message. The cause of the error is output in around the last fifteen lines of the log, so remove the cause of the error in accordance with the message and then re-execute the pgx_rcvall command.

   The following message displayed during recovery is output as part of normal operation of pgx_rcvall command (therefore the user does not need not be concerned).

   ```
   FATAL:  the database system is starting up
   ```

7. Start the instance

   Start the instance.

   Refer to "2.1.2 Using Server Commands" for information on how to start an instance.

8. Resume applications

   Resume applications that are using the database.

If failure occurred on the backup data storage disk

The procedure for recovering the backup data storage disk is described below.

There are two methods of taking action:

- Performing recovery while the instance is active
- Stopping the instance before performing recovery

The following table shows the different steps to be performed depending on whether you stop the instance.

No	Step	Instance stopped	
		No	Yes
1	Confirm that transaction log mirroring has stopped	Y	N
2	Stop output of archive logs	Y	N
3	Stop applications	N	Y
4	Stop the instance	N	Y
5	Recover the failed disk	Y	Y
6	Create a backup data storage destination directory	Y	Y
7	Resume output of archive logs	Y	N
8	Resume transaction log mirroring	Y	N
9	Start the instance	N	Y
10	Run backup	Y	Y
11	Resume applications	N	Y

Y: Required
N: Not required

The procedure is as follows:

If an instance has not been stopped

1. Confirm that transaction log mirroring has stopped

   Use the following SQL function to confirm that transaction log mirroring has stopped.

```
 postgres=# SELECT pgx_is_wal_multiplexing_paused();
 pgx_is_wal_multiplexing_paused
-------------------
 t
(1 row)
```

   If transaction log mirroring has not stopped, then stop it using the following SQL function.

```
postgres=# SELECT pgx_pause_wal_multiplexing();
LOG:  multiplexing of transaction log files has been stopped
pgx_pause_wal_multiplexing
---------------------------

(1 row)
```

2. Stop output of archive logs

   Transaction logs may accumulate during replacement of backup storage disk, and if the data storage disk or the transaction log storage disk becomes full, there is a risk that operations may not be able to continue.

   To prevent this, use the following methods to stop output of archive logs.

   - Changing archive_command

     Specify a command that will surely complete normally, such as "echo skipped archiving WAL file %f" or "/bin/true", so that archive logs will be regarded as having been output.

     If you specify echo, a message is output to the server log, so it may be used as a reference when you conduct investigations.

   - Reload the configuration file

     Execute the pg_ctl reload command or the pg_reload_conf SQL function to reload the configuration file.

   If you simply want to stop output of errors without the risk that operations will not be able to continue, specify an empty string ('') in archive_command and reload the configuration file.

3. Recover the failed disk

   Replace the disk, and then recover the volume configuration information.

4. Create a backup data storage destination

   Create a backup data storage destination.

   Example

   ```
   $ mkdir /database/inst1
   $ chown fsepuser:fsepuser /database/inst1
   $ chmod 700 /database/inst1
   ```

   Refer to "3.2.2 Using Server Commands" for information on how to create a backup data storage destination.

5. Resume output of archive logs

   Return the archive_command setting to its original value, and reload the configuration file.

6. Resume transaction log mirroring

   Execute the pgx_resume_wal_multiplexing SQL function.

   Example

   ```
   SELECT pgx_resume_wal_multiplexing()
   ```

7. Run backup

   Use the pgx_dmpall command to back up the database cluster.

   Specify the following value in the pgx_dmpall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

   Example

   ```
   > pgx_dmpall -D /database/inst1
   ```

If an instance has been stopped

1. Stop applications

   Stop applications that are using the database.

2. Stop the instance

   Stop the instance. Refer to "2.1.2 Using Server Commands" for details.

If the instance fails to stop, refer to "16.11 Actions in Response to Failure to Stop an Instance".

3. Recover the failed disk

   Replace the disk, and then recover the volume configuration information.

4. Create a backup data storage destination

   Create a backup data storage destination.

   Example

   ```
   # mkdir /backup/inst1
   # chown fsepuser:fsepuser /backup/inst1
   # chmod 700 /backup/inst1
   ```

   Refer to "3.2.2 Using Server Commands" for details.

5. Start the instance

   Start the instance. Refer to "2.1.2 Using Server Commands" for information on how to start an instance.

6. Run backup

   Use the pgx_dmpall command to back up the database cluster.

   Specify the following value in the pgx_dmpall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

   Example

   ```
   > pgx_dmpall -D /database/inst1
   ```

7. Resume applications

   Resume applications that are using the database.

See

- Refer to "pgx_rcvall" and "pgx_dmpall" in the Reference for information on the pgx_rcvall command and pgx_dmpall command.
- Refer to "Write Ahead Log" under "Server Administration" in the PostgreSQL Documentation for information on archive_command.
- Refer to "B.1 WAL Mirroring Control Functions" for information on pgx_resume_wal_multiplexing.

16.2 Recovering from Data Corruption

If data in a disk is logically corrupted and the database does not operate properly, you can recover the database cluster to its state at the time of backup.

There are two methods of recovery:

- 16.2.1 Using WebAdmin
- 16.2.2 Using the pgx_rcvall Command

Note

- Back up the database cluster after recovering it. Backup deletes obsolete archive logs (transaction logs copied to the backup data storage destination), freeing up disk space and reducing the recovery time.
- If you recover data to a point in the past, a new time series (database update history) will start from that recovery point. When recovery is complete, the recovery point is the latest point in the new time series. When you subsequently recover data to the latest state, the database update is re-executed on the new time series.

16.2.1 Using WebAdmin

If using WebAdmin, recover the data to the point immediately prior to data corruption by using the backup data.

Refer to "16.1.1 Using WebAdmin" for details.

16.2.2 Using the pgx_rcvall Command

Recover the database cluster by specifying in the pgx_rcvall command the date and time of the backup you want to read from. Then re-execute the transaction as required to recover the data.

Follow the procedure below to recover the data storage disk.

1. Stop applications

   Stop applications that are using the database.

2. Stop the instance

   Stop the instance. Refer to "2.1.2 Using Server Commands" for information on how to stop an instance.

   If the instance fails to stop, refer to "16.11 Actions in Response to Failure to Stop an Instance".

3. Confirm the backup date and time

   Execute the pgx_rcvall command to confirm the backup data saved in the backup data storage destination, and determine a date and time prior to data corruption.

   Specify the following values in the pgx_rcvall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
   - Specify the backup storage directory in the -B option.
   - The -l option displays the backup data information.

   Example

   ```
   > pgx_rcvall -D /database/inst1 -B /backup/inst1 -l
   Date                    Status          Dir
   2022-03-20 10:00:00     COMPLETE        /backup/inst1/2022-03-20_10-00-00
   ```

4. Recover the keystore, and enable automatic opening of the keystore

   When the data in the database has been encrypted, restore the keystore to its state at the time of the database backup. Configure automatic opening of the keystore as necessary.

5. Recover the database cluster

   Use the pgx_rcvall command to recover the database cluster.

   Specify the following values in the pgx_rcvall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
   - Specify the backup storage directory in the -B option.
   - Specify the recovery date and time in the -e option.

   Example

   In the following examples, "March 20, 2022 10:00:00" is specified as the recovery time.

   ```
   > pgx_rcvall -D /database/inst1 -B /backup/inst1 -e '2022-03-20 10:00:00'
   ```

Note

If recovery fails, remove the cause of the error in accordance with the displayed error message and then re-execute the pgx_rcvall command.

If the message "pgx_rcvall: an error occurred during recovery" is displayed, then the log recorded when recovery was executed is output after this message. The cause of the error is output in around the last fifteen lines of the log, so remove the cause of the error in accordance with the message and then re-execute the pgx_rcvall command.

The following message displayed during recovery is output as part of normal operation of pgx_rcvall command (therefore the user does not need not be concerned).

```
FATAL:  the database system is starting up
```

6. Start the instance

   Start the instance. Refer to "2.1.2 Using Server Commands" for information on how to start an instance.

   If necessary, re-execute transaction processing from the specified recovery time, and then resume database operations.

7. Resume applications

   Resume applications that are using the database.

See

Refer to "pgx_rcvall" in the Reference for information on the pgx_rcvall command.

16.3 Recovering from an Incorrect User Operation

This section describes how to recover database clusters when data has been corrupted due to erroneous user operations.

There are two methods of recovery:

- 16.3.1 Using WebAdmin
- 16.3.2 Using the pgx_rcvall Command

Note

- Back up the database cluster after recovering it. Backup deletes obsolete archive logs (transaction logs copied to the backup data storage destination), freeing up disk space and reducing the recovery time.
- If you recover data to a point in the past, a new time series (database update history) will start from that recovery point. When recovery is complete, the recovery point is the latest point in the new time series. When you subsequently recover data to the latest state, the database update is re-executed on the new time series.
- An effective restore point is one created on a time series for which you have made a backup. That is, if you recover data to a point in the past, you cannot use any restore points set after that recovery point. Therefore, once you manage to recover your target past data, make a backup.

16.3.1 Using WebAdmin

You can use WebAdmin to recover data to a backup point.

Note

Recovery operation cannot be performed on an instance that is part of a streaming replication cluster in standby mode.

If disk failure occurs on a standby instance, it may be necessary to delete and re-create the instance.

Recovery operation can be performed on an instance that is part of a streaming replication cluster in "Master" mode. If a recovery operation is performed on a master instance, it will break the replication cluster and streaming replication will stop between the master instance and all its standby instances. In such an event, the standby instances can be promoted to standalone instances or can be deleted and re-created.

Follow the procedure below to recover the data in the data storage disk.

1. Stop applications

   Stop applications that are using the database.

2. Stop the instance

   Stop the instance. Refer to "2.1.1 Using WebAdmin" for information on how to stop an instance.

3. Recover the keystore, and enable automatic opening of the keystore

   Do the following if the data in the database has been encrypted:

   - Restore the keystore to its state at the time of the database backup.
   - Enable automatic opening of the keystore.

4. Recover the database cluster

   Log in to WebAdmin, and in the [Instances] tab, select the instance to be recovered and click .

5. Recover to the backup point

   In the [Restore Instance] dialog box, click [Yes].

   Recovery is performed. An instance is automatically started when recovery is successful.

6. Resume database operations

   If necessary, re-execute transaction processing from the backup point to when an erroneous operation was performed, and then resume database operations.

16.3.2 Using the pgx_rcvall Command

The pgx_rcvall command recovers database clusters to the restore point created with the server command. Refer to "Setting a restore point" in "3.2.2 Using Server Commands" for information on how to create a restore point.

Follow the procedure below to recover the data in the data storage disk.

1. Stop applications

   Stop applications that are using the database.

2. Stop the instance

   Stop the instance. Refer to "2.1.2 Using Server Commands" for information on how to stop an instance.

   If the instance fails to stop, refer to "16.11 Actions in Response to Failure to Stop an Instance".

3. Confirm the restore point

   Execute the pgx_rcvall command to confirm the backup data saved in the backup data storage destination, and use a restore point recorded in an arbitrary file, as explained in "3.2.2 Using Server Commands", to determine a restore point prior to the erroneous operation.

   Specify the following values in the pgx_rcvall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
   - Specify the backup data storage destination in the -B option.
   - The -l option displays the backup data information.

Example

```
> pgx_rcvall -D /database/inst1 -B /backup/inst1 -l
Date                    Status          Dir
2022-03-01 10:00:00     COMPLETE        /backup/inst1/2022-03-01_10-00-00
```

4. Recover the keystore, and enable automatic opening of the keystore

   When the data in the database has been encrypted, restore the keystore to its state at the time of the database backup. Configure automatic opening of the keystore as necessary.

5. Recover the database cluster

   Use the pgx_rcvall command to recover the database cluster.

   Specify the following values in the pgx_rcvall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
   - Specify the backup data storage destination in the -B option.
   - The -n option recovers the data to the specified restore point.

   Example

   The following example executes the pgx_rcvall command with the restore point "batch_20220303_1".

```
> pgx_rcvall -D /database/inst1 -B /backup/inst1 -n batch_20220303_1
```

Note

If recovery fails, remove the cause of the error in accordance with the displayed error message and then re-execute the pgx_rcvall command.

If the message "pgx_rcvall: an error occurred during recovery" is displayed, then the log recorded when recovery was executed is output after this message. The cause of the error is output in around the last fifteen lines of the log, so remove the cause of the error in accordance with the message and then re-execute the pgx_rcvall command.

The following message displayed during recovery is output as part of normal operation of pgx_rcvall (therefore the user does not need not be concerned).

```
FATAL:  the database system is starting up
```

6. Start the instance

   Start the instance.

   Refer to "2.1.2 Using Server Commands" for information on how to start an instance.

7. Restart operation of the database

   If necessary, re-execute transaction processing from the specified recovery time to the point when an erroneous operation was performed, and then resume database operations.

See

Refer to "pgx_rcvall" in the Reference for information on the pgx_rcvall command.

16.4 Actions in Response to an Application Error

If there is a connection from a client that has been in the waiting state for an extended period, you can minimize performance degradation of the database by closing the problematic connection.

The following methods are available for identifying a connection to be closed:

- view(pg_stat_activity) (refer to "16.4.1 When using the view (pg_stat_activity)")
- ps command (refer to "16.4.2 Using the ps Command")
- pgAdmin (refer to "16.4.3 Using pgAdmin")

Use the system management function (pg_terminate_backend) to disconnect connections.

16.4.1 When using the view (pg_stat_activity)

When using the view (pg_stat_activity), follow the procedure below to close a connection.

1. Use psql command to connect to the postgres database.

```
> psql postgres
psql (<x>) (*1)
Type "help" for help.
```

*1: <x> indicates the PostgreSQL version on which this product is based.

2. Close connections from clients that have been in the waiting state for an extended period.

   Use pg_terminate_backend() to close connections that have been trying to connect for an extended period.

   However, when considering continued compatibility of applications, do not reference or use system catalogs and functions directly in SQL statements. Refer to " Notes on Application Compatibility" in the Application Development Guide for details.

   Example

   The following example closes connections where the client has been in the waiting state for at least 60 minutes.

```
select pid,usename,application_name,client_addr,pg_terminate_backend(pid) from pg_stat_activity
where backend_type = 'client backend' and state='idle in transaction' and current_timestamp >
cast(query_start + interval '60 minutes' as timestamp);
-[ RECORD 1 ]--------+---------------
pid                  | 4684
username             | fsepuser
application_name     | apl1
client_addr          | 192.11.11.1
pg_terminate_backend | t
```

See

- Refer to "System Administration Functions" under "The SQL Language" in the PostgreSQL Documentation for information on pg_terminate_backend.
- Refer to "Notes on Application Compatibility" in the Application Development Guide for information on how to maintain application compatibility.

16.4.2 Using the ps Command

Follow the procedure below to close a connection using a standard Unix tool (ps command).

1. Execute the ps command.
   Note that "<x>" indicates the product version.

```
> ps axwfo user,pid,ppid,tty,command | grep postgres
fsepuser 19174 18027 pts/1                \_ grep postgres
fsepuser 20517     1 ?      /opt/fsepv<x>server64/bin/postgres -D /disk01/data
fsepuser 20518 20517 ?       \_ postgres: logger
fsepuser 20520 20517 ?       \_ postgres: checkpointer
fsepuser 20521 20517 ?       \_ postgres: background writer
fsepuser 20522 20517 ?       \_ postgres: walwriter
```

```
fsepuser 20523 20517 ?        \_ postgres: autovacuum launcher
fsepuser 20524 20517 ?        \_ postgres: archiver
fsepuser 20525 20517 ?        \_ postgres: logical replication launcher
fsepuser 18673 20517 ?        \_ postgres: fsepuser postgres 192.168.100.1(49448) idle
fsepuser 16643 20517 ?        \_ postgres: fsepuser db01 192.168.100.11(49449) UPDATE waiting
fsepuser 16644 20517 ?        \_ postgres: fsepuser db01 192.168.100.12(49450) idle in transaction
```

Process ID 16643 may be a connection that was established a considerable time ago by the UPDATE statement, or a connection that has occupied resources (waiting).

2. Close connections from clients that have been in the waiting state for an extended period.

   Use pg_terminate_backend() to close the connection with the process ID identified in step 1 above.

   The example below disconnects the process with ID 16643.

   However, when considering continued compatibility of applications, do not reference or use system catalogs and functions directly in SQL statements.

```
postgres=# SELECT pg_terminate_backend (16643);
 pg_terminate_backend
----------------------
t
(1 row)
```

See

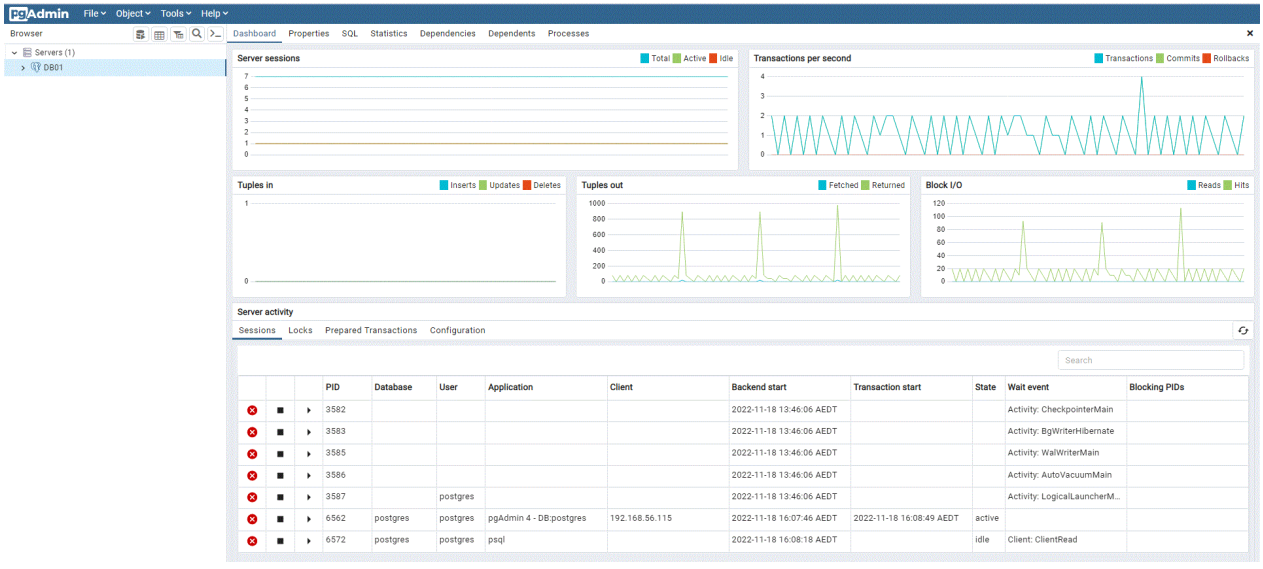
- Refer to "System Administration Functions" under "The SQL Language" in the PostgreSQL Documentation for information on pg_terminate_backend.
- Refer to "Notes on Application Compatibility" in the Application Development Guide for information on how to maintain application compatibility.

16.4.3 Using pgAdmin

If using pgAdmin, follow the procedure below to close connections.

1. In the [Browser] pane, click the relevant database server.
2. Close the client connections that have been in a wait state for an extended period.

   Click the [Dashboard] tab. In the [Server activity] section, select the connections that have been in an "idle" or "idle in transaction" state for an extended period. For each of these connections, click to close the session.



16.5 Actions in Response to an Access Error

If access is denied, grant privileges allowing the instance administrator to operate the following directories, and then re-execute the operation. Also, refer to the event log and the server log, and confirm that the file system has not been mounted as read-only due to a disk error. If the file system has been mounted as read-only, mount it properly and then re-execute the operation.

- Data storage destination
- Tablespace storage destination
- Transaction log storage destination
- Backup data storage destination

Refer to "Preparing Directories to Deploy Resources" under "Setup" in the Installation and Setup Guide for Server for information on the privileges required for the directory.

16.6 Actions in Response to Insufficient Space on the Data Storage Destination

If the data storage destination runs out of space, check if the disk contains any unnecessary files and delete them so that operations can continue.

If deleting unnecessary files does not solve the problem, you must migrate data to a disk with larger capacity.

There are two methods of migrating data:

- 16.6.1 Using a Tablespace
- 16.6.2 Replacing the Disk with a Larger Capacity Disk

16.6.1 Using a Tablespace

Fujitsu Enterprise Postgres enables you to use a tablespace to change the storage destination of database objects, such as tables and indexes, to a different disk.

The procedure is as follows:

1. Create a tablespace

   Use the CREATE TABLESPACE command to create a new tablespace in the prepared disk.

2. Modify the tablespace

   Use the ALTER TABLE command to modify tables for the newly defined tablespace.

Refer to "SQL Commands" under "Reference" in the PostgreSQL Documentation for information on the CREATE TABLESPACE command and ALTER TABLE command.

16.6.2 Replacing the Disk with a Larger Capacity Disk

Before replacing the disk with a larger capacity disk, migrate resources at the data storage destination using the backup and recovery features.

There are two methods of performing backup and recovery:

- 16.6.2.1 Using WebAdmin

- 16.6.2.2 Using Server Commands

The following sections describe procedures that use each of these methods to replace the disk and migrate resources at the data storage destination.

Note

- Before replacing the disk, stop applications and instances that are using the database.
- It is recommended that you back up the database cluster following recovery. Backup deletes obsolete archive logs (transaction logs copied to the backup data storage destination), freeing up disk space and reducing the recovery time.

16.6.2.1 Using WebAdmin

Follow the procedure below to replace the disk and migrate resources at the data storage destination by using WebAdmin.

1. Back up files

   If the disk at the data storage destination contains any required files, back up the files. It is not necessary to back up the data storage destination.

2. Stop applications

   Stop applications that are using the database.

3. Back up the database cluster

   Back up the latest resources at the data storage destination. Refer to "3.2.1 Using WebAdmin" for details.

4. Stop the instance

   Stop the instance. Refer to "2.1.1 Using WebAdmin" for information on how to stop an instance.

5. Replace with a larger capacity disk

   Replace the disk. Then, recover the volume configuration information.

6. Recover the database cluster

   Log in to WebAdmin, and perform recovery operations. Refer to steps 4 ("Create a tablespace directory ") to 7 ("Run recovery") under "If failure occurred in the data storage disk or the transaction log storage disk" in "16.1.1 Using WebAdmin" for information on the procedure. An instance is automatically started when recovery is successful.

7. Resume applications

   Resume applications that are using the database.

8. Restore the files

   Restore the files backed up in step 1.

16.6.2.2 Using Server Commands

Follow the procedure below to replace the disk and migrate resources at the data storage destination by using server commands.

1. Back up files

   If the disk at the data storage destination contains any required files, back up the files. It is not necessary to back up the data storage destination.

2. Stop applications

   Stop applications that are using the database.

3. Back up the database cluster

   Back up the latest resources at the data storage destination. Refer to "3.2.2 Using Server Commands" for details.

4. Stop the instance

   After backup is complete, stop the instance. Refer to "2.1.2 Using Server Commands" for information on how to stop an instance.

   If the instance fails to stop, refer to "16.11 Actions in Response to Failure to Stop an Instance".

5. Replace with a larger capacity disk

   Replace the disk. Then, recover the volume configuration information.

6. Create a data storage destination

   Create a data storage destination. If a tablespace was defined, also create a directory for it.

   Example

   ```
   $ mkdir /database/inst1
   $ chown fsepuser:fsepuser /database/inst1
   $ chmod 700 /database/inst1
   ```

7. Recover the keystore, and enable automatic opening of the keystore

   When the data in the database has been encrypted, restore the keystore to its state at the time of the database backup. Configure automatic opening of the keystore as necessary.

8. Recover the database cluster

   Use the pgx_rcvall command to recover the database cluster.

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
   - Specify the backup storage directory in the -B option.

   Example

   ```
   > pgx_rcvall -D /database/inst1 -B /backup/inst1
   ```

   Note

   If recovery fails, remove the cause of the error in accordance with the displayed error message and then re-execute the pgx_rcvall command.

   If the message "pgx_rcvall: an error occurred during recovery" is displayed, then the log recorded when recovery was executed is output after this message. The cause of the error is output in around the last fifteen lines of the log, so remove the cause of the error in accordance with the message and then re-execute the pgx_rcvall command.

   The following message displayed during recovery is output as part of normal operation of pgx_rcvall (therefore the user does not need not be concerned).

   ```
   FATAL:  the database system is starting up
   ```

   See

   Refer to "pgx_rcvall" in the Reference for information on the pgx_rcvall command.

9. Start the instance

   Start the instance.

   Refer to "2.1.2 Using Server Commands" for information on how to start an instance.

10. Resume applications

    Resume applications that are using the database.

11. Restore files

    Restore the files backed up in step 1.

16.7 Actions in Response to Insufficient Space on the Backup Data Storage Destination

If space runs out on the backup data storage destination, check if the disk contains any unnecessary files and delete them, and then make a backup as required.

If deleting unnecessary files does not solve the problem, take the following action:

- 16.7.1 Temporarily Saving Backup Data
- 16.7.2 Replacing the Disk with a Larger Capacity Disk

16.7.1 Temporarily Saving Backup Data

This method involves temporarily moving backup data to a different directory, saving it there, and securing disk space on the backup data storage destination so that a backup can be made normally.

Use this method if you need time to prepare a larger capacity disk.

If space runs out on the backup data storage destination, archive logs can no longer be stored in the backup data storage destination. As a result, transaction logs continue to accumulate in the data storage destination or the transaction log storage destination.

If action is not taken soon, the transaction log storage destination will become full, and operations may not be able to continue.

To prevent this, secure space in the backup data storage destination, so that archive logs can be stored.

There are two methods of taking action:

- 16.7.1.1 Using WebAdmin
- 16.7.1.2 Using Server Commands

16.7.1.1 Using WebAdmin

Follow the procedure below to recover the backup data storage disk.

1. Temporarily save backup data

   Move backup data to a different directory and temporarily save it, and secure space in the backup data storage destination directory.

   The reason for saving the backup data is so that the data in the data storage destination can be recovered even if it is corrupted before you perform recovery. If there is no disk at the save destination and you consider that there is no risk of corruption at the data storage destination, delete the backup data.

   The following example saves backup data from the backup data storage destination directory (/backup/inst1) under /mnt/usb/backup.

   Example

   ```
   > mkdir /mnt/usb/backup/
   > mv /backup/inst1/* /mnt/usb/backup/
   ```

2. Back up the database cluster

   Back up the latest resources at the data storage destination. Refer to "3.2.1 Using WebAdmin" for details.

3. Delete temporarily saved backup data

   If backup completes normally, the temporarily saved backup data becomes unnecessary and is deleted.

   The following example deletes backup data that was temporarily saved in /mnt/usb.

   Example

   ```
   > rm -rf /mnt/usb/backup
   ```

16.7.1.2 Using Server Commands

The following describes the procedure for recovering the backup storage disk.

There are two methods of taking action:

- Performing recovery while the instance is active
- Stopping the instance before performing recovery

The following table shows the different steps to be performed depending on whether you stop the instance.

No	Step	Instance stopped	
		No	Yes
1	Stop transaction log mirroring	Y	N
2	Stop output of archive logs	Y	N
3	Stop applications	N	Y
4	Stop the instance	N	Y
5	Temporarily save backup data	Y	Y
6	Resume output of archive logs	Y	N
7	Resume transaction log mirroring	Y	N
8	Start an instance	N	Y
9	Run backup	Y	Y
10	Resume applications	N	Y
11	Delete temporarily saved backup data	Y	Y

Y: Required
N: Not required

The procedure is as follows:

Performing recovery while the instance is active

1. Stop transaction log mirroring

   Stop transaction log mirroring.

   ```
   postgres=# SELECT pgx_pause_wal_multiplexing();
   LOG:  multiplexing of transaction log files has been stopped
   pgx_pause_wal_multiplexing
   ---------------------------

   (1 row)
   ```

2. Stop output of archive logs

   Transaction logs may accumulate during replacement of backup storage disk, and if the data storage disk or the transaction log storage disk becomes full, there is a risk that operations may not be able to continue.

   To prevent this, use the following methods to stop output of archive logs.

   - Changing the archive_command parameter

     Specify a command that will surely complete normally, such as "echo skipped archiving WAL file %f" or "/bin/true", so that archive logs will be regarded as having been output.

     If you specify echo, a message is output to the server log, so it may be used as a reference when you conduct investigations.

- Reloading the configuration file

  Run the pg_ctl reload command or the pg_reload_conf SQL function.

If you simply want to stop output of errors without the risk that operations will not be able to continue, specify an empty string ('') in archive_command and reload the configuration file.

3. Temporarily save backup data

   Move backup data to a different directory and temporarily save it, and secure space in the backup data storage destination directory.

   The reason for saving the backup data is so that the data in the data storage destination can be recovered even if it is corrupted before you perform the next step. If there is no disk at the save destination and you consider that there is no risk of corruption at the data storage destination, delete the backup data.

   The following example saves backup data from the backup data storage destination directory (/backup/inst1) under /mnt/usb/backup.

   Example

   ```
   > mkdir /mnt/usb/backup/
   > mv /backup/inst1/* /mnt/usb/backup/
   ```

4. Resume output of archive logs

   Return the archive_command setting to its original value, and reload the configuration file.

5. Resume transaction log mirroring

   Execute the pgx_resume_wal_multiplexing SQL function.

   Example

   ```
   SELECT pgx_resume_wal_multiplexing()
   ```

6. Run backup

   Use the pgx_dmpall command to back up the database cluster.

   Specify the following option in the pgx_dmpall command:

   - Specify the directory of the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

     Example

     ```
     > pgx_dmpall -D /database/inst1
     ```

7. Delete temporarily saved backup data

   If backup completes normally, the temporarily saved backup data becomes unnecessary and is deleted.

   The following example deletes backup data that was temporarily saved in /mnt/usb.

   Example

   ```
   > rm -rf /mnt/usb/backup
   ```

If an instance has been stopped

1. Stop applications

   Stop applications that are using the database.

2. Stop the instance

   Stop the instance. Refer to "2.1.2 Using Server Commands" for details.

   If the instance fails to stop, refer to "16.11 Actions in Response to Failure to Stop an Instance".

3. Temporarily save backup data

   Move backup data to a different directory and temporarily save it, and secure space in the backup data storage destination directory.

The reason for saving the backup data is so that the data in the data storage destination can be recovered even if it is corrupted before you perform recovery. If there is no disk at the save destination and you consider that there is no risk of corruption at the data storage destination, delete the backup data.

The following example saves backup data from the backup data storage destination directory (/backup/inst1) under /mnt/usb/backup.

Example

```
> mkdir /mnt/usb/backup/
> mv /backup/inst1/* /mnt/usb/backup/
```

4. Start the instance

   Start the instance. Refer to "2.1.2 Using Server Commands" for information on how to start an instance.

5. Run backup

   Use the pgx_dmpall command to back up the database cluster.

   Specify the following value in the pgx_dmpall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

     Example

     ```
     > pgx_dmpall -D /database/inst1
     ```

6. Resume applications

   Resume applications that are using the database.

7. Delete temporarily saved backup data

   If backup completes normally, the temporarily saved backup data becomes unnecessary and is deleted.

   The following example deletes backup data that was temporarily saved in /mnt/usb.

   Example

   ```
   > rm -rf /mnt/usb/backup
   ```

See

- Refer to "pgx_rcvall" and "pgx_dmpall" in the Reference for information on the pgx_rcvall command and pgx_dmpall command.
- Refer to "Write Ahead Log" under "Server Administration" in the PostgreSQL Documentation for information on archive_command.
- Refer to "B.1 WAL Mirroring Control Functions" for information on the pgx_is_wal_multiplexing_paused and pgx_resume_wal_multiplexing.

16.7.2 Replacing the Disk with a Larger Capacity Disk

This method involves replacing the disk at the backup data storage destination with a larger capacity disk, so that it does not run out of free space again. After replacing the disk, back up data to obtain a proper backup.

There are two methods of performing backup:

- 16.7.2.1 Using WebAdmin
- 16.7.2.2 Using Server Commands

Note

Before replacing the disk, stop applications that are using the database.

16.7.2.1 Using WebAdmin

Follow the procedure below to recover the backup storage disk.

1. Back up files

   If the disk at the backup data storage destination contains any required files, back up the files. It is not necessary to back up the backup data storage destination.

2. Temporarily save backup data

   Save the backup data to a different directory.

   The reason for saving the backup data is so that the data in the data storage destination can be recovered even if it is corrupted before you perform the next step. If there is no disk at the save destination and you consider that there is no risk of corruption at the data storage destination, delete the backup data.

   The following example saves backup data from the backup data storage destination directory (/backup/inst1) under /mnt/usb/backup.

   Example

   ```
   > mkdir /mnt/usb/backup/
   > mv /backup/inst1/* /mnt/usb/backup/
   ```

3. Replace with a larger capacity disk

   Replace the disk. Then, recover the volume configuration information.

4. Run backup

   Log in to WebAdmin, and perform recovery operations. Refer to steps 2 ("Recover the backup data") and 3 ("Run backup") under "If failure occurred on the backup storage disk" in "16.1.1 Using WebAdmin".

5. Restore files

   Restore the files backed up in step 1.

6. Delete temporarily saved backup data

   If backup completes normally, the temporarily saved backup data becomes unnecessary and is deleted.

   The following example deletes backup data that was temporarily saved in /mnt/usb.

   Example

   ```
   > rm -rf /mnt/usb/backup
   ```

16.7.2.2 Using Server Commands

The procedure for recovering the backup data storage disk is described below.

There are two methods of taking action:

- Performing recovery while the instance is active
- Stopping the instance before performing recovery

The following table shows the different steps to be performed depending on whether you stop the instance.

No	Step	Instance stopped	
		No	Yes
1	Back up files	Y	Y
2	Temporarily save backup data	Y	Y
3	Confirm that transaction log mirroring has stopped	Y	N
4	Stop output of archive logs	Y	N
5	Stop applications	N	Y

No	Step	Instance stopped	
		No	Yes
6	Stop the instance	N	Y
7	Replace with a larger capacity disk	Y	Y
8	Create a backup storage directory	Y	Y
9	Resume output of archive logs	Y	N
10	Resume transaction log mirroring	Y	N
11	Start the instance	N	Y
12	Run backup	Y	Y
13	Resume applications	N	Y
14	Restore files	Y	Y
15	Delete temporarily saved backup data	Y	Y

Y: Required
N: Not required

The procedure is as follows:

If an instance has not been stopped

1. Back up files

   If the disk at the backup data storage destination contains any required files, back up the files. It is not necessary to back up the backup data storage destination.

2. Temporarily save backup data

   Save the backup data to a different directory.

   The reason for saving the backup data is so that the data in the data storage destination can be recovered even if it is corrupted before you perform the next step. If there is no disk at the save destination and you consider that there is no risk of corruption at the data storage destination, delete the backup data.

   The following example saves backup data from the backup data storage destination directory (/backup/inst1) under /mnt/usb/backup.

   #### Example

   ```
   > mkdir /mnt/usb/backup/
   > mv /backup/inst1/* /mnt/usb/backup/
   ```

3. Confirm that transaction log mirroring has stopped

   Use the following SQL function to confirm that transaction log mirroring has stopped.

   ```
    postgres=# SELECT pgx_is_wal_multiplexing_paused();
    pgx_is_wal_multiplexing_paused
   -------------------
    t
   (1 row)
   ```

   If transaction log mirroring has not stopped, then stop it using the following SQL function.

   ```
   postgres=# SELECT pgx_pause_wal_multiplexing();
   LOG:  multiplexing of transaction log files has been stopped
   pgx_pause_wal_multiplexing
   ----------------------------
   ```

```
(1 row)
```

4. Stop output of archive logs

   Transaction logs may accumulate during replacement of backup storage disk, and if the data storage destination disk or the transaction log storage destination disk becomes full, there is a risk that operations may not be able to continue.

   To prevent this, use the following methods to stop output of archive logs.

   - Changing the archive_command parameter

     Specify a command that will surely complete normally, such as "echo skipped archiving WAL file %f" or "/bin/true", so that archive logs will be regarded as having been output.

     If you specify echo, a message is output to the server log, so it may be used as a reference when you conduct investigations.

   - Reloading the configuration file

     Run the pg_ctl reload command or the pg_reload_conf SQL function.

   If you simply want to stop output of errors without the risk that operations will not be able to continue, specify an empty string ('') in archive_command and reload the configuration file.

5. Replace with a larger capacity disk

   Replace the disk. Then, recover the volume configuration information.

6. Create a backup data storage destination

   Create a backup data storage destination.

   Example

   ```
   # mkdir /backup/inst1
   # chown fsepuser:fsepuser /backup/inst1
   # chmod 700 /backup/inst1
   ```

   Refer to "3.2.2 Using Server Commands" for details.

7. Resume output of archive logs

   Return the archive_command setting to its original value, and reload the configuration file.

8. Resume transaction log mirroring

   Execute the pgx_resume_wal_multiplexing SQL function.

   Example

   ```
   SELECT pgx_resume_wal_multiplexing()
   ```

9. Run backup

   Use the pgx_dmpall command to back up the database cluster.

   Specify the following value in the pgx_dmpall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

   Example

   ```
   > pgx_dmpall -D /database/inst1
   ```

10. Restore files

    Restore the files backed up in step 1.

11. Delete temporarily saved backup data

    If backup completes normally, the temporarily saved backup data becomes unnecessary and is deleted.

    The following example deletes backup data that was temporarily saved in /mnt/usb.

Example

```
> rm -rf /mnt/usb/backup
```

If an instance has been stopped

1. Back up files

   If the disk at the backup data storage destination contains any required files, back up the files. It is not necessary to back up the backup data storage destination.

2. Temporarily save backup data

   Save the backup data to a different directory.

   The reason for saving the backup data is so that the data in the data storage destination can be recovered even if it is corrupted before you perform the next step. If there is no disk at the save destination and you consider that there is no risk of corruption at the data storage destination, delete the backup data.

   The following example saves backup data from the backup data storage destination directory (/backup/inst1) under /mnt/usb/backup.

   Example

   ```
   > mkdir /mnt/usb/backup/
   > mv /backup/inst1/* /mnt/usb/backup/
   ```

3. Stop applications

   Stop applications that are using the database.

4. Stop the instance

   Stop the instance. Refer to "2.1.2 Using Server Commands" for information on how to stop an instance.

   If the instance fails to stop, refer to "16.11 Actions in Response to Failure to Stop an Instance".

5. Replace with a larger capacity disk

   Replace the disk. Then, recover the volume configuration information.

6. Create a backup data storage destination

   Create a backup data storage destination.

   Example

   ```
   # mkdir /backup/inst1
   # chown fsepuser:fsepuser /backup/inst1
   # chmod 700 /backup/inst1
   ```

   Refer to "3.2.2 Using Server Commands" for details.

7. Start the instance

   Start the instance. Refer to "2.1.2 Using Server Commands" for information on how to start an instance.

8. Run backup

   Use the pgx_dmpall command to back up the database cluster.

   Specify the following value in the pgx_dmpall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

   Example

   ```
   > pgx_dmpall -D /database/inst1
   ```

9. Resume applications

   Resume applications that are using the database.

10. Restore files

    Restore the files backed up in step 1.

11. Delete temporarily saved backup data

    If backup completes normally, the temporarily saved backup data becomes unnecessary and is deleted.

    The following example deletes backup data that was temporarily saved in /mnt/usb.

    Example

    ```
    > rm -rf /mnt/usb/backup
    ```

See

- Refer to "pgx_rcvall" and "pgx_dmpall" in the Reference for information on the pgx_rcvall command and pgx_dmpall command.
- Refer to "Write Ahead Log" under "Server Administration" in the PostgreSQL Documentation for information on archive_command.
- Refer to "B.1 WAL Mirroring Control Functions" for information on the pgx_is_wal_multiplexing_paused and pgx_resume_wal_multiplexing.

16.8 Actions in Response to Insufficient Space on the Transaction Log Storage Destination

If the transaction log storage destination runs out of space, check if the disk contains any unnecessary files and delete them so that operations can continue.

If deleting unnecessary files does not solve the problem, you must migrate data to a disk with larger capacity.

16.8.1 Replacing the Disk with a Larger Capacity Disk

Before replacing the disk with a larger capacity disk, migrate resources at the transaction log storage destination using the backup and recovery features.

There are two methods of performing backup and recovery:

- 16.8.1.1 Using WebAdmin
- 16.8.1.2 Using Server Commands

The following sections describe procedures that use each of these methods to replace the disk and migrate resources at the transaction log storage destination.

Note

- Before replacing the disk, stop applications that are using the database.
- It is recommended that you back up the database cluster following recovery. Backup deletes obsolete archive logs (transaction logs copied to the backup data storage destination), freeing up disk space and reducing the recovery time.

16.8.1.1 Using WebAdmin

Follow the procedure below to replace the disk and migrate resources at the transaction log storage destination by using WebAdmin.

1. Back up files

   If the disk at the transaction log storage destination contains any required files, back up the files. It is not necessary to back up the transaction log storage destination.

2. Back up the database cluster

   Back up the latest data storage destination resources and transaction log storage destination resources (refer to "3.2.1 Using WebAdmin" for details).

3. Stop applications

   Stop applications that are using the database.

4. Stop the instance

   Stop the instance. Refer to "2.1.1 Using WebAdmin" for information on how to stop an instance. WebAdmin automatically stops instances if recovery of the database cluster is performed without stopping the instance.

5. Replace with a larger capacity disk

   Replace the disk. Then, recover the volume configuration information.

6. Create a tablespace directory

   If a tablespace was defined after backing up, create a directory for it.

7. Recover the keystore, and enable automatic opening of the keystore

   Do the following if the data in the database has been encrypted:

   - Restore the keystore to its state at the time of the database backup.
   - Enable automatic opening of the keystore.

8. Recover the database cluster

   Log in to WebAdmin, and perform recovery operations. Refer to steps 4 ("Create a tablespace directory ") to 7 ("Run Recovery") under " If failure occurred in the data storage disk or the transaction log storage disk " in "16.1.1 Using WebAdmin" for information on the procedure. An instance is automatically started when recovery is successful.

9. Resume applications

   Resume applications that are using the database.

10. Restore files

    Restore the files backed up in step 1.

16.8.1.2 Using Server Commands

Follow the procedure below to replace the disk and migrate resources at the transaction log storage destination by using server commands.

1. Back up files

   If the disk at the transaction log storage destination contains any required files, back up the files. It is not necessary to back up the transaction log storage destination.

2. Back up the database cluster

   Use server commands to back up the latest data storage destination resources and transaction log storage destination resources. Refer to "3.2.2 Using Server Commands" for information on how to perform backup.

3. Stop applications

   Stop applications that are using the database.

4. Stop the instance

   After backup is complete, stop the instance. Refer to "2.1.2 Using Server Commands" for information on how to stop an instance.

   If the instance fails to stop, refer to "16.11 Actions in Response to Failure to Stop an Instance".

5. Replace with a larger capacity disk

   Replace the disk. Then, recover the volume configuration information.

6. Create a transaction log storage destination

   Create a transaction log storage destination. If a tablespace was defined, also create a directory for it.

   Example

   ```
   # mkdir /tranlog/inst1
   # chown fsepuser:fsepuser /tranlog/inst1
   # chmod 700 /tranlog/inst1
   ```

7. Recover the keystore, and enable automatic opening of the keystore

   When the data in the database has been encrypted, restore the keystore to its state at the time of the database backup. Configure automatic opening of the keystore as necessary.

8. Recover the database cluster

   Use the pgx_rcvall command to recover the database cluster.

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
   - Specify the backup storage directory in the -B option.

   Example

   ```
   > pgx_rcvall -D /database/inst1 -B /backup/inst1
   ```

   Note

   If recovery fails, remove the cause of the error in accordance with the displayed error message and then re-execute the pgx_rcvall command.

   If the message "pgx_rcvall: an error occurred during recovery" is displayed, then the log recorded when recovery was executed is output after this message. The cause of the error is output in around the last fifteen lines of the log, so remove the cause of the error in accordance with the message and then re-execute the pgx_rcvall command.

   The following message displayed during recovery is output as part of normal operation of pgx_rcvall command (therefore the user does not need not be concerned).

   ```
   FATAL: the database system is starting up
   ```

   See

   Refer to "pgx_rcvall" in the Reference for information on the pgx_rcvall command.

9. Start the instance

   Start the instance.

   Refer to "2.1.2 Using Server Commands" for information on how to start an instance.

10. Resume applications

    Resume applications that are using the database.

11. Restore files

    Restore the files backed up in step 1.

16.9 Errors in More Than One Storage Disk

If an error occurs in the storage destination disks or resources are corrupted, determine the cause of the error from system logs and server logs and remove the cause.

If errors occur in either of the following combinations, you cannot recover the database.

Recreate the instance, and rebuild the runtime environment.

Data storage disk	Transaction log storage disk	Backup data storage disk
Error	-	Error
-	Error	Error

See

Refer to "Setup" in the Installation and Setup Guide for Server for information on how to create an instance and build the runtime environment.

16.10 Actions in Response to Instance Startup Failure

If an instance fails to start, refer to the system log and the server log, and determine the cause of the failure.

If using WebAdmin, remove the cause of the error. Then, click [Solution] and [Recheck the status] and confirm that the instance is in the normal state.

The following sections describe common causes of errors and the actions to take.

16.10.1 Errors in the Configuration File

If you have directly edited the configuration file using a text editor or changed the settings using WebAdmin, refer to the system log and the server log, confirm that no messages relating to the files below have been output.

- postgresql.conf
- pg_hba.conf

See

Refer to the following for information on the parameters in the configuration file:

- "Configuring Parameters" in the Installation and Setup Guide for Server
- "Appendix A Parameters"
- "Server Configuration" and "Client Authentication" under "Server Administration" in the PostgreSQL Documentation

16.10.2 Errors Caused by Power Failure or Mounting Issues

If mounting is cancelled after restarting the server, for example, because the disk device for each storage destination disk was not turned on, or because automatic mounting has not been set, then starting an instance will fail.

Refer to "16.14.2 Errors Caused by Power Failure or Mounting Issues", and take actions accordingly.

16.10.3 Other Errors

This section describes the recovery procedure to be used if you cannot take any action or the instance cannot start even after you have referred to the system log and the server log.

There are two methods of recovery:

- 16.10.3.1 Using WebAdmin
- 16.10.3.2 Using Server Commands

Note that recovery will not be possible if there is an error at the backup data storage destination. If the problem cannot be resolved, contact Fujitsu technical support.

16.10.3.1 Using WebAdmin

Follow the procedure below to perform recovery.

1. Delete the data storage destination directory and the transaction log storage destination directory

   Back up the data storage destination directory and the transaction log storage destination directory before deleting them.

2. Reconfirm the status

   Log in to WebAdmin, and in the [Instances] tab, click [Solution] for the error message.

   Click [Recheck the status] to reconfirm the storage destination resources.

3. Run recovery

   Restore the database cluster after WebAdmin detects an error.

   Refer to "16.2.1 Using WebAdmin" for details.

16.10.3.2 Using Server Commands

Follow the procedure below to recover the database.

1. Delete the data storage destination directory and the transaction log storage destination directory

   Save the data storage destination directory and the transaction log storage destination directory, and then delete them.

2. Execute recovery

   Use the pgx_rcvall command to recover the database cluster.

   Refer to "16.2.2 Using the pgx_rcvall Command" for details.

16.11 Actions in Response to Failure to Stop an Instance

If an instance fails to stop, refer to the system log and the server log, and determine the cause of the failure.

If the instance cannot stop despite taking action, perform the following operation to stop the instance.

There are two methods of recovery:

- 16.11.1 Using WebAdmin
- 16.11.2 Using Server Commands

16.11.1 Using WebAdmin

In the [Instances] tab, click and select the Fast stop mode or the Immediate stop mode to stop the instance. Forcibly terminate the server process from WebAdmin if the instance cannot be stopped.

Refer to "2.1.1 Using WebAdmin" for information on the stop modes.

16.11.2 Using Server Commands

There are three methods:

- Stopping the Instance Using the Fast Mode

  If backup is in progress, then terminate it, roll back all executing transactions, forcibly close client connections, and then stop the instance.

- Stopping the Instance Using the Immediate Mode

  Forcibly terminate the instance immediately. A crash recovery is run when the instance is restarted.

- Forcibly Stopping the Server Process

  Reliably stops the server process when the other methods are unsuccessful.

16.11.2.1 Stopping the Instance Using the Fast Mode

Specify "-m fast" in the pg_ctl command to stop the instance.

If the instance fails to stop when you use this method, stop the instance as described in "16.11.2.2 Stopping the Instance Using the Immediate Mode" or "16.11.2.3 Forcibly Stopping the Server Process".

Example

```
> pg_ctl stop -D /database/inst1 -m fast
```

16.11.2.2 Stopping the Instance Using the Immediate Mode

Specify "-m immediate " in the pg_ctl command to stop the instance.

If the instance fails to stop when you use this method, stop the instance as described in "16.11.2.3 Forcibly Stopping the Server Process".

Example

```
> pg_ctl stop -D /database/inst1 -m immediate
```

16.11.2.3 Forcibly Stopping the Server Process

If both the Fast mode and the Immediate mode fail to stop the instance, use the kill command or the kill parameter of the pg_ctl command to forcibly stop the server process.

The procedure is as follows:

1. Execute the ps command
   Note that "<*x*>" indicates the product version.

   ```
   > ps axwfo user,pid,ppid,tty,command | grep postgres
   fsepuser 19174 18027 pts/1                \_ grep postgres
   fsepuser 20517     1 ?        /opt/fsepv<x>server64/bin/postgres -D /database/inst1
   fsepuser 20518 20517 ?         \_ postgres: logger
   fsepuser 20520 20517 ?         \_ postgres: checkpointer
   fsepuser 20521 20517 ?         \_ postgres: background writer
   fsepuser 20522 20517 ?         \_ postgres: walwriter
   fsepuser 20523 20517 ?         \_ postgres: autovacuum launcher
   fsepuser 20524 20517 ?         \_ postgres: archiver
   fsepuser 20525 20517 ?         \_ postgres: logical replication launcher
   ```

   The process ID (20517) indicates the server process.

2. Forcibly stop the server process

   As instance manager, forcibly stop the server process.

Using the pg_ctl command

```
> pg_ctl kill SIGQUIT 20517
```

Using the kill command

```
> kill -s SIGQUIT 20517
```

16.12 Actions in Response to Failure to Create a Streaming Replication Standby Instance

When creating a streaming replication standby instance using WebAdmin, if the instance creation fails, refer to the system log and the server log, and determine the cause of the failure.

When an error occurs in the creation of the standby instance using WebAdmin, it is unlikely that the partially created standby instance can be resumed to complete the operation.

In such a scenario, fix the cause of the error, delete the partially created standby instance, and then create a new standby instance. This recommendation is based on the following assumptions:

- As the instance is yet to be created completely, there are no applications connecting to the database.
- The standby instance is in error state and is not running.
- There are no backups for the standby instance and as a result, it cannot be recovered.

See

Refer to "Deleting Instances" in the Installation and Setup Guide for details on how to delete an instance.

16.13 Actions in Response to Error in a Distributed Transaction

If a system failure (such as server failure) occurs in an application that uses distributed transactions (such as .NET TransactionScope), then transactions may be changed to the in-doubt state.

At that point, resources accessed by the transaction will be locked, and rendered unusable by other transactions.

The following describes how to check for in-doubt transactions, and how to resolve them.

How to check for in-doubt transactions

The following shows how to check for them:

If the server fails

1. An in-doubt transaction will have occurred if a message similar to the one below is output to the log when the server is restarted.

   Example

   ```
   LOG: Restoring prepared transaction 2103.
   ```

2. Refer to system view pg_prepared_xacts to obtain information about the prepared transaction.

   If the transaction identifier of the prepared transaction in the list (in the transaction column of pg_prepared_xacts) is the same as the identifier of the in-doubt transaction obtained from the log output when the server was restarted, then that row is the information about the in-doubt transaction.

   Example

   ```
   postgres=# select * from pg_prepared_xacts;
    transaction |   gid     |   prepared |  owner   | database
   -------------+-----------+------------+----------+----------
   ```

```
2103 | 374cc221-f6dc-4b73-9d62-d4fec9b430cd | 2022-03-06 16:28:48.471+08 | postgres |
postgres (1 row)
```

Information about the in-doubt transaction is output to the row with the transaction ID 2103 in the transaction column.

If the client fails

If there are no clients connected and there is a prepared transaction in pg_prepared_xacts, then you can determine that the transaction is in the in-doubt state.

If at least one client is connected and there is a prepared transaction in pg_prepared_xacts, you cannot determine whether there is a transaction in the in-doubt state. In this case, use the following query to determine the in-doubt transaction from the acquired database name, user name, the time PREPARE TRANSACTION was executed, and the information about the table name accessed.

```
select gid,x.database,owner,prepared,l.relation::regclass as relation from pg_prepared_xacts x
left join pg_locks l on l.virtualtransaction = '-1/'||x.transaction and l.locktype='relation';
```

If it still cannot be determined from this information, wait a few moments and then check pg_prepared_xacts again.

If there is a transaction that has continued since the last time you checked, then it is likely that it is the one in the in-doubt state.

Point

As you can see from the explanations in this section, there is no one way to definitively determine in-doubt transactions.

Consider collecting other supplementary information (for example, logging on the client) or performing other operations (for example, allocating database users per job).

How to resolve in-doubt transactions

From the system view pg_prepared_xacts mentioned above, obtain the global transaction identifier (in the gid column of pg_prepared_xacts) for the in-doubt transaction, and issue either a ROLLBACK PREPARED statement or COMMIT PREPARED statement to resolve the in-doubt transaction.

Example

- Rolling back in-doubt transactions

```
postgres=# rollback prepared '374cc221-f6dc-4b73-9d62-d4fec9b430cd';
ROLLBACK PREPARED
```

- Committing in-doubt transactions

```
postgres=# commit prepared '374cc221-f6dc-4b73-9d62-d4fec9b430cd';
COMMIT PREPARED
```

16.14 I/O Errors Other than Disk Failure

Even if a disk is not defective, the same input-output error messages, as those generated when the disk is defective, may be output.

A few examples of such errors are given below. The appropriate action for each error is explained respectively.

- 16.14.1 Network Error with an External Disk
- 16.14.2 Errors Caused by Power Failure or Mounting Issues

16.14.1 Network Error with an External Disk

This is an error that occurs in the network path to/from an external disk.

Determine the cause of the error by checking the information in the system log and the server log, the disk access LED, network wiring, and network card status. Take appropriate action to remove the cause of the error, for example, replace problematic devices.

16.14.2 Errors Caused by Power Failure or Mounting Issues

These are errors that occur when the disk device is not turned on, automatic mounting of the disk was not set, or mounting was accidentally cancelled.

In this case, check the information in the system log and the server log, the disk access LED, and whether the disk is mounted correctly. If problems are detected, take appropriate action.

If mounting has been cancelled, it is possible that mounting was accidentally cancelled, or automatic mounting at the time of starting the operating system is not set. In this case, set the mounting to be performed automatically.

16.15 Anomaly Detection and Resolution

The following operations performed via the command line interface will result in an anomaly in WebAdmin:

- Changes to the port and backup_destination parameters in postgresql.conf
- Changes to Mirroring Controller configuration of cluster replication added via WebAdmin

This section describes when WebAdmin checks for such anomalies, and what takes place when an anomaly is detected.

16.15.1 Port Number and Backup Storage Path Anomalies

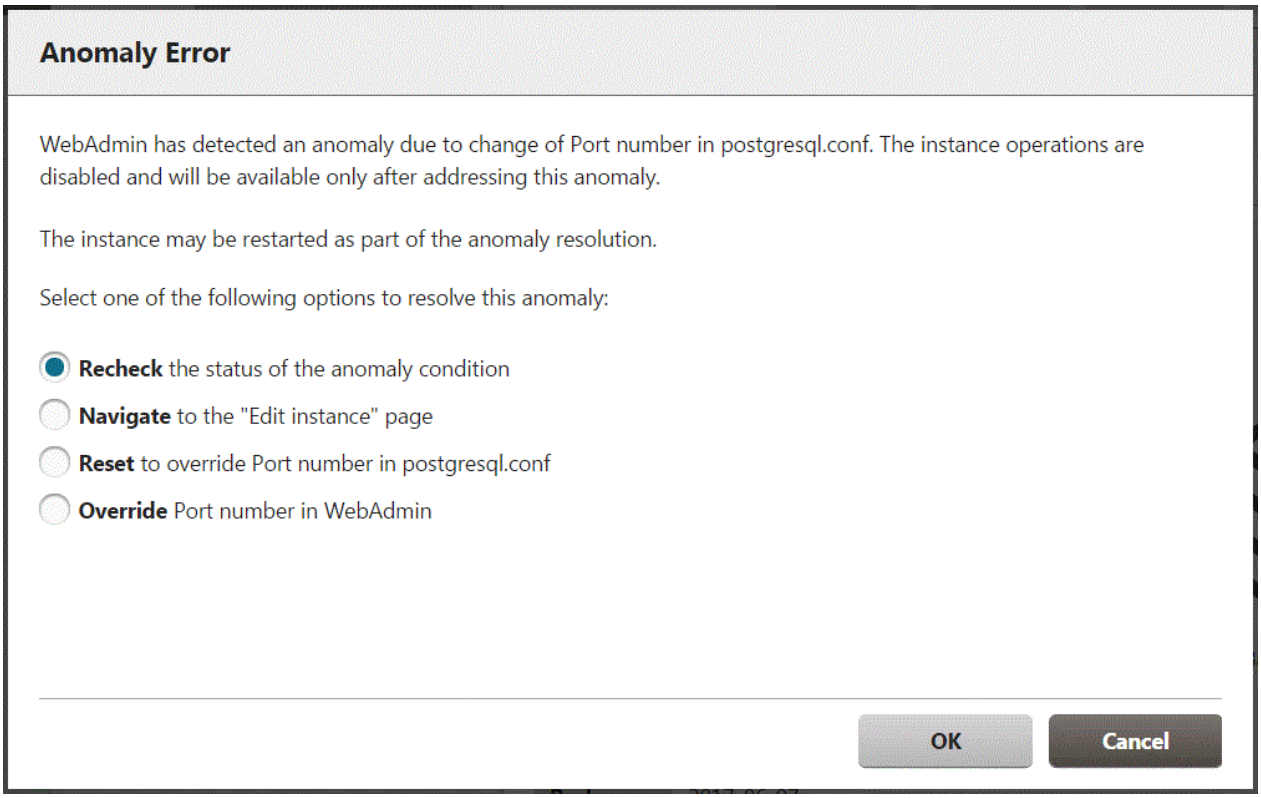
An anomaly occurs when the value of [Port number] and/or [Backup storage path] in WebAdmin is different from the value of its corresponding parameter in postgresql.conf - port and backup_destination, respectively.

WebAdmin checks for anomalies when an instance is selected for viewing or any instance operation is performed. Anomalies will be identified for the selected instance only.

The following occurs when an anomaly is detected in port number and/or backup storage path:

- All instance operation buttons are disabled, except for "Edit instance", "Refresh instance", and "Delete Mirroring Controller"
- A red error status indicator is displayed on the instance icon
- For an anomaly specific to backup storage path, a red error status indicator is displayed on the [Backup storage] disk icon, and [Backup storage status] is set to "Error"
- The message, "WebAdmin has detected an anomaly with...", is displayed in the [Message] section along with an associated [Solution] button

Click [Solution]. The [Anomaly Error] dialog box is displayed.



Select the required option, click [OK], and then resolve the anomaly error.

Refer to "Editing instance information" in the Installation and Setup Guide for Server for information on the [Edit instance] page.

Note

Critical errors encountered during anomaly resolution will be displayed, however, rollback of the instance to its previous state is not supported.

16.15.2 Mirroring Controller Anomalies

The following conditions will cause a Mirroring Controller anomaly:

- The Mirroring Controller management folder or configuration files have been deleted
- The permissions to the Mirroring Controller management folder or configuration files have been changed such that:
  - The instance administrator's access to Mirroring Controller configuration is denied
  - Users other than an instance administrator have access privileges to Mirroring Controller configuration files

WebAdmin checks for anomalies when Mirroring Controller status check is performed.

The following occurs when a Mirroring Controller anomaly is detected:

- All Mirroring Controller functionality is disabled for the replication cluster, except for "Delete Mirroring Controller"
- [Mirroring Controller status] is set to "Error"
- Either of the following messages is displayed in the [Message] section

  "Failed to access the Mirroring Controller management folder or configuration files '*path*'. Mirroring Controller functionality has been disabled. Consider deleting Mirroring Controller and adding it again."

  "Failed to find the Mirroring Controller management folder or configuration files '*path*'. Mirroring Controller functionality has been disabled. Consider deleting Mirroring Controller and adding it again."

Appendix A Parameters

This appendix describes the parameters to be set in the postgresql.conf file of Fujitsu Enterprise Postgres.

The postgresql.conf file is located in the data storage destination.

Information

The maximum value that can be expressed as a 4-byte signed integer changes according to the operating system. Follow the definition of the operating system in use.

- core_directory (string)

  This parameter specifies the directory where the corefile is to be output. If this parameter is omitted, the data storage destination is used by default. This parameter can only be set when specified on starting an instance. It cannot be changed dynamically, while an instance is active.

- core_contents (string)

  This parameter specifies the contents to be included in the corefile.

  - full: Outputs all contents of the server process memory to the corefile.
  - none: Does not output a corefile.
  - minimum: Outputs only non-shared memory server processes to the corefile. This reduces the size of the corefile. However, in some cases, this file may not contain sufficient information for examining the factor that caused the corefile to be output.

  If this parameter is omitted, "minimum" is used by default. This parameter can only be set when specified on starting an instance. It cannot be changed dynamically, while an instance is active.

- keystore_location (string)

  This parameter specifies the directory that stores the keystore file. Specify a different location from other database clusters. This parameter can only be set when specified on starting an instance. It cannot be changed dynamically, while an instance is active.

  Cannot be specified with the tde_kms.kms_conninfo_file parameter.

- tde_kms.plugin_path (string)

  When connecting to the key management system using a plug-in, specify the directory that stores the plug-in with an absolute path. Only database administrators should be able to store plugins in this directory. It can only be set by specifying a parameter when starting the instance.

- tde_kms.enable_shared_dek (boolean)

  Enables or disables sharing of data encryption keys between backend processes for each encrypted tablespace in transparent data encryption. Default is off. It can only be set by specifying a parameter when starting the instance.

- tde_kms.max_shared_dek (numerical value)

  Specify the maximum number of shared data encryption keys when sharing data encryption keys per tablespace in transparent data encryption. Default is 1000. It can only be set by specifying a parameter when starting the instance.

- tde_kms.kms_conninfo_file (string)

  When using the key management service as a key store, specifies the file that contains the connection information for the key management system. Cannot be specified with the keystore_location parameter.

  Create a connection information file for the key management system in one of the following format:

```
kmip    kms-name    address    port    auth-method    [auth-options]
custom  kms-name    plugin-name      [plugin-options]   [extra-args]
```

  Specify one of the following methods to connect to the key management system.

- kmip

  Access this key management system using the KMIP protocol. Take out and use the encryption key in this key management system.

- custom

  Access the key management system using a plugin module. The encryption/decryption process is done inside the plugin or inside the key management system. Fujitsu Enterprise Postgres does not use cryptographic keys directly.

For type kmip

- kms-name

  The key management system name assigned to the key management system and specified when declaring the master encryption key or opening the keystore. The name of the key management system must be unique within this file. The key management system name must be a string of no more than 63 characters beginning with a-z, consisting of a-z, a number (0 -9), and an underscore. Upper and lower case letters are the same.

- address

  Specifies the host name or IP address of the key management service.

- port

  Specifies the port number on which the key management service listens for services.

- auth-method

  Specifies the authentication method for the key management service.

- auth-options

  The auth-method is followed by the authentication method options. You can specify multiple options in a name = value field.

Authentication method when using a key management service of type kmip

cert

A certificate is used to authenticate the KMIP server and the client, Fujitsu Enterprise Postgres, to each other. The auth-options can be.

- sslcert

  Specifies the file name of the client certificate. The corresponding format is PEM format.

- sslkey

  Specifies the file name of the private key used for the client certificate. The corresponding format is PEM format. If you choose to encrypt the file with a passphrase, use a passphrase that is no more than 1023 bytes long.

- sslkeypassphrase-obf

  Specifies the file that contains the obfuscated passphrase for the private key file specified by sslkey. This option allows the keystore to be opened automatically when the server starts. The pgx _ keystore command creates obfuscated files. It can be omitted.

- sslrootcert

  Specifies the file name of the SSL Certificate Authority certificate. The corresponding format is PEM format. Used to verify the server certificate of the connection destination.

Example

```
kmip   mykmipsvr  mykmipsvr.example.com   5696   cert   sslcert=postgres.crt
sslkey=postgres.key sslrootcert=root.crt
```

Note

cert authentication does not verify that the server you are connecting to is the same server you are trying to connect to. Any server using a server certificate that is signed with the certificate of the certificate authority specified in sslrootcert is considered the correct destination. To avoid problems with this behavior, consider using your own CA or self-signed certificate for the KMIP server.

For type custom

- kms-name

  The key management system name given to the key management system, specified when declaring a master encryption key or opening a keystore. The name of the key management system must be unique within this file. The key management system name must be a string of up to 63 characters starting with a-z and consisting of a-z, digits (0-9) and underscores. Uppercase and lowercase letters are considered the same.

- plugin-name

  Specify the name of the plugin. Use the executable file with the same name as the plugin name in the directory specified by tde_kms.plugin_path as the plugin module.

- plugin-options

  Specify other options for the plugin. Multiple options can be specified in name=value format fields. You can specify the following option names.

  - kms-secret-obf

    Specify the file containing the obfuscated KMS secret when enabling automatic opening of the keystore using transparent data encryption. The obfuscated file is created with the pgx_keystore command. The obfuscated file contents are decrypted by Fujitsu Enterprise Postgres and passed to the plugin. Can be omitted if you are not using transparent data encryption to enable automatic opening of the keystore.

- extra-args

  Specify additional arguments to pass to the specified shell command in the form arg=value. If you specify multiple extra-args, the values are passed to the shell command in that order.

Example

```
custom   mykms  mykms  arg=--profile arg=user1
```

- tablespace_encryption_algorithm (string)

  This parameter specifies the encryption algorithm for tablespaces that will be created. Valid values are "AES128", "AES256", and "none". If you specify "none", encryption is not performed. The default value is "none". To perform encryption, it is recommended that you specify "AES256". Only superusers can change this setting.

- backup_destination (string)

  This parameter specifies the absolute path of the directory where pgx_dmpall will store the backup data. Specify a different location from other database clusters. This parameter can only be set when specified on starting an instance. It cannot be changed dynamically, while an instance is active.

  Place this directory on a different disk from the data directory to be backed up and the tablespace directory. Ensure that users do not store arbitrary files in this directory, because the contents of this directory are managed by the database system.

- search_path (string)

  When using the SUBSTR function compatible with Oracle databases, set "oracle" and "pg_catalog" in the search_path parameter. You must specify "oracle" before "pg_catalog".

Example

```
search_path = '"$user", public, oracle, pg_catalog'
```

Information

- The search_path feature specifies the priority of the schema search path. The SUBSTR function in Oracle database is defined in the oracle schema.
- Refer to "Statement Behavior" under "Server Administration" in the PostgreSQL Documentation for information on search_path.

- track_waits (string)

  This parameter enables collection of statistics for pgx_stat_lwlock and pgx_stat_latch.

  - on: Enables collection of statistics.
  - off: Disables collection of statistics.

  If this parameter is omitted, "on" is assumed.

  Only superusers can change this setting.

- track_sql (string)

  This parameter enables collection of statistics for pgx_stat_sql.

  - on: Enables collection of statistics.
  - off: Disables collection of statistics.

  If this parameter is omitted, "on" is assumed.

  Only superusers can change this setting.

Parameters for the in-memory feature

- reserve_buffer_ratio (numerical value)

  This parameter specifies the proportion of shared memory to be used for a stable buffer table.

  - Minimum value: 0
  - Maximum value: 80

  If this parameter is omitted, 0 will be used.

- vci.cost_threshold (numerical value)

  This parameter specifies the lowest cost that selects an execution plan that uses a VCI. If the cost of the best execution plan that does not use a VCI is lower than this value, that execution plan will be selected.

  - Minimum value: 0
  - Maximum value: Maximum value that can be expressed as a 4-byte signed integer

  If this parameter is omitted or a value outside this range is specified, 18000 will be used.

- vci.control_max_workers (numerical value)

  This parameter specifies the number of background workers that manage VCI. The number of workers for the entire instance is limited by max_worker_processes, so add the value specified here to max_worker_processes.

  - Minimum value: 1
  - Maximum value: 8388607

  If this parameter is omitted or a value outside this range is specified, 8 will be used.

- vci.enable (string)

  This parameter enables or disables VCI.

  - on: Enables VCI.
  - off: Disables VCI.

  If this parameter is omitted, "on" will be used.

- vci.log_query (string)

  This parameter enables or disables log output when VCI is not used due to insufficient memory specified by vci.max_local_ros.

  - on: Enables log output.
  - off: Disables log output.

  If this parameter is omitted, "off" will be used.

- vci.maintenance_work_mem (numerical value)

  This parameter specifies the maximum memory size used for maintenance of VCI (when executing CREATE INDEX, for example).

  - Minimum value: 1 MB
  - Maximum value: Maximum value that can be expressed as a 4-byte signed integer

  If this parameter is omitted or a value outside this range is specified, 256 MB will be used.

- vci.max_local_ros (numerical value)

  This parameter specifies the maximum memory size used for VCI scan.

  - Minimum value: 64 MB
  - Maximum value: Maximum value that can be expressed as a 4-byte signed integer

  If this parameter is omitted or a value outside this range is specified, 64 MB will be used.

- vci.max_parallel_degree (numerical value)

  This parameter specifies the maximum number of background workers used for parallel scan. The number of workers for the entire instance is limited by max_worker_processes, so add the value specified here to max_worker_processes.

  A value from -8388607 to 8388607 can be specified.

  - Integer (1 or greater): Parallel scan is performed using the specified degree of parallelism.
  - 0: Stops the parallel scan process.
  - Negative number: The specified value minus the maximum number of CPUs obtained from the environment is used as the degree of parallelism and parallel scan is performed.

  If this parameter is omitted or a value outside this range is specified, 0 will be used.

- vci.shared_work_mem (numerical value)

  This parameter specifies the maximum memory size used for VCI parallel scan.

  - Minimum value: 32 MB
  - Maximum value: Maximum value that can be expressed as a 4-byte signed integer

  If this parameter is omitted or a value outside this range is specified, 1 GB will be used.

Parameters for the Global Meta Cache feature

- pgx_global_metacache (numerical value)

  Specifies the memory size of the GMC area.

  Specify a value calculated by the formula below.

A value lower than the calculated value will still work, but the meta cache may not be able to fit into the GMC area.

In this case, the system will discard the meta cache it thinks it is no longer needed, but if it is needed again, the meta cache will need to be expanded and will not perform well.

If the value is less than 10 MB and is set to a nonzero value that disables the feature, the database startup fails because the Global Meta Cache feature cannot operate.

A setting of 0 disables the Global Meta Cache feature. The default is 0.

Changing this setting requires restarting the database.

```
Size of GMC area
 = Max(10MB,
     (All user table x 0.4 KB
      + All user Indexes x 0.3 KB
      + All user columns x 0.8 KB) x 1.5 (*1) )

*1) Safety Factor (1.5)
This value takes into account the case where both GMC before and after the change temporarily exist
at the same time in shared memory when the table definition is changed or the row of the system
catalog is changed.
```

- track_gmc (string)

  This parameter enables collection of statistics for pgx_stat_gmc.

  - on: Enables collection of statistics.
  - off: Disables collection of statistics.

  If this parameter is omitted, "on" is used.

  Only superusers can change this setting.

Parameters for the Local Meta Cache Limit feature

- pgx_catalog_cache_max_size(numerical value)

  Specifies the maximum amount of memory that the backend process should use as the catalog cache.

  You can enable catalog cache deletion by setting it to 8 KB or more.

  A setting of 0 disables the catalog cache removal. The default is 0.

  If no units are specified, they are treated as KB.

  - Minimum value: 8KB
  - Maximum value: Maximum value that can be expressed as a 4-byte signed integer

  When calculating the parameter settings, the factors that determine the cache size are calculated as the number of tables, the number of indexes, and the number of columns. What is kept as a catalog cache or relation cache also includes objects such as databases, roles, or procedures, but these are small compared to the above factors and do not need to be factored into them. It also includes a calculation method for pgx_relacion_cache_max_size because the given memory is distributed between the catalog cache and the relation cache.

The calculation method here assumes that all backends have similar access and that the transaction also has access to a similar number of resources. If you have a small number of singular backends or transactions, consider excluding them as errors.

1. Determine how much memory a backend process can use. Decide by subtracting the memory size required by the entire system such as the database cache from the installed memory and dividing the rest by the number of connections.
2. For best performance, use the following formula to calculate the total memory size of the catalog cache when the backend holds the catalog cache for all resources accessed during its lifetime.

The amount of memory varies depending on whether Global Meta Cache is enabled or disabled. Enabling Global Meta Cache reduces the amount of memory required because most of the cache is located on shared memory.

```
When Global Meta Cache is enabled:
  (Number of tables to access + Number of indexes to access + Number of columns to access)
      × 0.1KB × 1.5 (*1)
When Global Meta Cache is disabled:
  { Number of tables to access × 0.5KB(pg_class tuple size)
      + Number of indexes to access × 0.5KB(pg_index tuple size)
      + Number of columns to access × 1.0KB(pg_statistic tuple size)} × 1.5 (*1)

*1) Safety Factor (1.5)
The system catalog contains columns with variable-length types. For example, the tuple size in
pg_class is a constant value multiplied by the number of tables, while relname in pg_class is
variable length data.
It is not practical to calculate every definition in detail, so we added 50% to the above
formula.
```

3. In the same way as in 2., calculate the relation cache using the following formula.

   ```
   (1.4KB × Number of tables to access + 2.4KB × Number of indexes to access) × 1.5 (*1)

   *1) Safety Factor (1.5)
   The relation cache is structured to facilitate the use of table and index definitions, holds
   pointers to various objects, and is sized to include them. It is variable length because the
   type of object allocated by the table definition and its size change. Since it is not
   realistic to calculate for all definitions, 50% is added.
   ```

4. If the value of 1. ≥ the value of 2. + the value of 3., the backend process can keep all caches to the extent allowed, so there is no need to limit the caches. If you want to cap for safety, set the value of 2. to pgx_catalog_cache_max_size and the value of 3. to pgx_relation_cache_max_size.
5. If the value of 1. < the value of 2. + the value of 3. then you need to limit the cache. However, this parameter does not limit the size of the cache used by a transaction. Therefore, take the following steps.
6. Calculate the catalog cache used by a transaction using the formula in 2.
7. Calculate the relation cache used by a transaction using the formula in 3.
8. If the value of 1. < the value of 6. + the value of 7., then the value of 1. needs to be increased. In other words, in some cases, it may be necessary to increase the installed memory or reduce the number of connections.
9. If the value of 1. ≥ the value of 6. + the value of 7., the condition of 1. can be satisfied by limiting the cache with this parameter. Divide the value of 1. by the ratio of 2. and 3. and set it as a parameter. Set the value distributed to 2. to pgx_catalog_cache_max_size and the value distributed to 3. to pgx_relation_cache_max_size.
10. The value calculated in 9. is a provisional value. If you cannot meet your target performance, first try to shift the focus of allocation to the relation cache. This is because when executing SQL, the relation cache generated based on the catalog cache is mainly referenced, so it is advantageous to leave a large amount of relation cache. If the performance is still not satisfied, adjust the parameters by referring to "14.1.4 Performance Impact and Parameter Tuning of the Local Meta Cache Limit Feature".

Note

Be careful when partitioning the table.

The cached definition changes depending on whether the parent table is specified in the SQL statement or the child table is specified. In particular, note that if you specify a parent table, the definitions of all child tables are cached. This is because when you specify a parent table in an SQL statement, you need to know the definitions of all the child tables in order to determine which child table will contain the desired data. Note that the column information of the parent table is not cached.

When specifying the parent table:

```
Number of tables to access = Number of parent tables to access + Number of defined child tables
Number of columns = Number of defined columns x number of defined child tables
```

When specifying the child table directly:

```
Number of tables to access = Number of child tables actually accessed
Number of columns = Number of defined columns x number of child tables actually accessed
```

Example)

Suppose the parent table T (1 index, 3 columns) is split from child tables T1 to T5 (1 index, 3 columns, respectively). If the parent table T is specified in SQL, when the child tables that contain the data to be queried are limited to T1 and T2, and when accessing the data using the indexes defined by T1 and T2, calculate as follows.

```
Number of tables = 1(parent table) + 5(child table) = 6
Number of indexes = 2 (index to access)
Number of columns = 3 (number of columns) x 5 (child table) = 15
```

If you specify child tables T1 and T2 in SQL and use the indexes defined on T1 and T2 when accessing data, the calculation is as follows.

```
Number of tables = 2(child table)
Number of indexes = 2 (index to access)
Number of columns = 3 (number of columns) x 2 (child table) = 6
```

- pgx_relation_cache_max_size(numerical value)

  Specifies the maximum amount of memory that the backend process should use as the relation cache.

  You can enable catalog cache deletion by setting it to 8 KB or more.

  A setting of 0 disables the relation cache removal. The default is 0.

  If no units are specified, they are treated as KB.

  - Minimum value: 8KB
  - Maximum value: Maximum value that can be expressed as a 4-byte signed integer

  For the calculation method for parameter setting, refer to the calculation method of pgx_catalog_cache_max_size.

- pgx_cache_hit_log_interval(numerical value)

  Specifies the time interval to output a message indicating the cache reference status for each backend process.

  When the transaction ends, if the time set in this parameter has elapsed since the previous message was output, the message is output.

  If set to 0, a message will be output each time the transaction ends.

  Setting -1 disables the output. The default value is 10min.

  If no units are specified, they are treated as ms.

  Even if pgx_catalog_cache_max_size and pgx_relation_cache_max_size are disabled, the message output of the corresponding cache will be invalid.

  Immediately after connecting to the server, a small transaction occurs before the request from the user application, such as for user authentication. Since it is meaningless to know the hit rate for these, a message will be output at the end of the transaction that started after the time set in this parameter has elapsed after connecting to the server.

  For the same reason, setting a small value such as 0 may result in a message being printed at the end of such a small transaction.

  You can check which transaction the message corresponds to from the information output at the beginning.

  This information depends on the setting of the parameter log_line_prefix.

  - Minimum value: 0
  - Maximum value: 2147483647ms

See

Refer to "Server Configuration" under "Server Administration" in the PostgreSQL Documentation for information on other postgresql.conf parameters.

Appendix B System Administration Functions

This appendix describes the system administration functions of Fujitsu Enterprise Postgres.

See

Refer to "System Administration Functions" under "The SQL Language" in the PostgreSQL Documentation for information on other system administration functions.

B.1 WAL Mirroring Control Functions

The following table lists the functions that can be used for backup and recovery based on WAL mirroring.

Table B.1 WAL mirroring control functions

Name	Return type	Description
pgx_pause_wal_multiplexing()	void	Stops WAL multiplexing
pgx_resume_wal_multiplexing()	void	Resumes WAL multiplexing
pgx_is_wal_multiplexing_paused()	boolean	Returns true if WAL multiplexing has stopped

If WAL multiplexing has not been configured, these functions return an error. Setting the backup_destination parameter in postgresql.conf configures WAL multiplexing.

Only superusers can execute these functions.

B.2 Transparent Data Encryption Control Functions

The following table lists the functions that can be used for transparent data encryption.

Table B.2 Transparent data encryption control functions

Name	Return type	Description
pgx_open_keystore(*passphrase*) pgx_open_keystore(sslpassphrase => text)	void	Opens the keystore
pgx_set_master_key(*passphrase*)	void	Sets the master encryption key
pgx_declare_external_master_key(kms_name => text, key_id => text, sslpassphrase => text)	void	To set an encryption key existing in a key management system as a master encryption key for transparent data encryption.
pgx_set_keystore_passphrase(*oldPassphrase*, *newPassphrase*)	void	Changes the keystore passphrase

B.2.1 pgx_open_keystore

pgx_open_keystore opens the keystore.

Only superusers can execute this function. Also, this function cannot be executed within a transaction block.

File-based keystores:

The pgx_open_keystore function uses the specified passphrase to open the keystore. When the keystore is opened, the master encryption key is loaded into the database server memory. In this way, you can access the encrypted data and create encrypted tablespaces. If the keystore is already open, this function returns an error.

Using the key management system as a keystore

pgx_open_keystore makes available (opens a keystore) a master encryption key on a key management system that has already been declared for use. The keystore cannot be opened unless it has been declared to use a master encryption key.

If the keystore is already open, use the credentials you entered to reconnect to the key management system.

Specify the authentication information for connecting to the key management system. Arguments must be specified in naming notation. The information you pass in the argument depends on the key management system you use.

If the key management system information file specifies an obfuscated credentials file, the file is recreated with the new credentials.

Using the key management service of type kmip

The following arguments are specified in naming notation.

- sslpassphrase text

  Specifies the passphrase of the client certificate private key file when connecting to the KMIP server. This can be omitted if no passphrase is set in the private key file.

Using the key management service of type custom

The following arguments are specified in naming notation.

- kms_secret text

  Confidential information passed to the plugin. It can be omitted if it is not necessary for using the key management system. Whether or not it can be omitted depends on the implementation of the plugin.

Example

To specify the passphrase mykmippassphrase for the client certificate private key file in naming notation:

```
SELECT pgx_open_keystore( sslpassphrase => 'mykmippassphrase' );
```

B.2.2 pgx_set_master_key

The pgx_set_master_key function generates a master encryption key and stores it in the file-based keystore.

If the keystore does not exist, this function creates a keystore. If the keystore already exists, this function modifies the master encryption key. If the keystore has not been opened, this function opens it.

The passphrase is a string of 8 to 200 bytes.

Only superusers can execute this function. Also, this function cannot be executed within a transaction block. Processing is not affected by whether the keystore is open.

B.2.3 pgx_declare_external_master_key

pgx_declare_external_master_key declares the use of an encryption key that exists in the key management system as the master encryption key for transparent data encryption. If the master encryption key already exists, change the master encryption key. If the master encryption key already exists, the keystore must be open.

The argument specifies information that identifies the master encryption key. Arguments must be specified in naming notation. The information you pass in the argument depends on the key management system you use.

This function can only be executed by superuser. Also, you cannot execute this function within a transaction block.

This function is available if you have installed the extension 'tde_kms'.

The following arguments are specified in naming notation:

- kms_name text

  Specify the key management system name specified in the key management system connection information file. Required.

- key_id text

  Specify the key ID assigned to the encryption key. Cannot be omitted.

- sslpassphrase text

  Specify the passphrase of the client certificate private key file when connecting to the KMIP server. This can be omitted if the private key file does not have a passphrase. Ignored if the key management system type specified by kms_name is not kmip.

- kms_secret text

  Confidential information passed to the plugin. It can be omitted if it is not necessary for using the key management system. Whether or not it can be omitted depends on the implementation of the plugin. Ignored if the key management system type specified by kms_name is not custom.

Example

```
SELECT pgx_declare_external_master_key( kms_name => 'mykmipsvr', key_id =>
'a0eebc99-9c0b-0000-0000-000000000000', sslpassphrase => 'mykmippassphrase' );
```

B.2.4 pgx_set_keystore_passphrase

The pgx_set_keystore_passphrase function changes the file-based keystore passphrase.

Specify the current passphrase in *oldPassphrase*, and a new passphrase in *newPassphrase*.

The passphrase is a string of 8 to 200 bytes.

Only superusers can execute this function. Also, this function cannot be executed within a transaction block. Processing is not affected by whether the keystore is open.

B.3 Data Masking Control Functions

The table below lists the functions that can be used for data masking.

Table B.3 Data masking control functions

Name	Return type	Description
pgx_alter_confidential_policy	boolean	Changes masking policies
pgx_create_confidential_policy	boolean	Creates masking policies
pgx_drop_confidential_policy	boolean	Deletes masking policies
pgx_enable_confidential_policy	boolean	Enables or disables masking policies
pgx_update_confidential_values	boolean	Changes replacement characters when full masking is specified for masking type

B.3.1 pgx_alter_confidential_policy

Description

Changes masking policies

Format

The format varies depending on the content to be changed. The format is shown below.

- Common format

```
common_arg:
[schema_name     := 'schemaName',]
```

```
table_name    := 'tableName',
policy_name     := 'policyName'
```

- Add a masking target to a masking policy

```
pgx_alter_confidential_policy(
commonArg,
[action        := 'ADD_COLUMN', ]
column_name    := 'colName'
[, function_type    := 'FULL'] |
[, function_type    := 'PARTIAL', partialOpt] |
[, function_type    := 'REGEXP', regexpOpt]
)


partialOpt:
function_parameters    := 'maskingFmt'


regexpOpt:
regexp_pattern        := 'regexpPattern',
regexp_replacement    := 'regexpReplacementChar',
[, regexp_flags        := 'regexpFlags']
```

- Delete a masking target from a masking policy

```
pgx_alter_confidential_policy(
commonArg,
action        := 'DROP_COLUMN',
column_name    := 'colName'
)
```

- Change the masking condition

```
pgx_alter_confidential_policy(
commonArg,
action        := 'MODIFY_EXPRESSION',
expression    := 'expr'
)
```

- Change the content of a masking policy set for a masking target

```
pgx_alter_confidential_policy(
commonArg,
action        := 'MODIFY_COLUMN',
column_name    := 'colName'
[, function_type    := 'FULL'] |
[, function_type    := 'PARTIAL', partialOpt] |
[, function_type    := 'REGEXP', regexpOpt]
)


partialOpt:
function_parameters    := 'maskingFmt'


regexpOpt:
regexp_pattern        := 'regexpPattern',
regexp_replacement        := 'regexpReplacementChar',
[, regexp_flags        := 'regexpFlags']
```

- Change the masking policy description

```
pgx_alter_confidential_policy(
commonArg,
```

```
action        := 'SET_POLICY_DESCRIPTION',
policy_description    := 'policyDesc'
)
```

- Change the masking target description

```
pgx_alter_confidential_policy(
commonArg,
action        := 'SET_COLUMN_DESCRIPTION',
column_name    := 'colName',
column_description    := 'colDesc'
)
```

Argument

The argument varies depending on the content to be changed. Details are as follows.

- Common arguments

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	schema_name	varchar(63)	Schema name of table for which a masking policy is applied	'public'
	table_name	varchar(63)	Name of table for which a masking policy is applied	Mandatory
	policy_name	varchar(63)	Masking policy name	Mandatory

- Add a masking target to a masking policy

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	action	varchar(63)	'ADD_COLUMN'	'ADD_COLUMN'
	column_name	varchar(63)	Masking target name	Mandatory
	function_type	varchar(63)	Masking type - 'FULL': Full masking - 'PARTIAL': Partial masking - 'REGEXP': Regular expression masking	'FULL'
Partial masking	function_parameters	varchar(1024)	Masking format for partial masking	Mandatory
Regular expression masking	regexp_pattern	varchar(1024)	Search pattern for regular expression masking	Mandatory
	regexp_replacement	varchar(1024)	Replacement character/string for regular expression masking	Mandatory
	regexp_flags	varchar(20)	Regular expression flags	NULL

- Delete a masking target from a masking policy

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	action	varchar(63)	'DROP_COLUMN'	Mandatory
	column_name	varchar(63)	Masking target name	Mandatory

- Change the masking condition

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	action	varchar(63)	'MODIFY_EXPRESSION'	Mandatory
	expression	varchar(1024)	Masking condition to be changed	Mandatory

- Change the content of a masking policy set for a masking target

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	action	varchar(63)	'MODIFY_COLUMN'	Mandatory
	column_name	varchar(63)	Masking target name	Mandatory
	function_type	varchar(63)	Masking type - 'FULL': Full masking - 'PARTIAL': Partial masking - 'REGEXP': Regular expression masking	'FULL'
Partial masking	function_parameters	varchar(1024)	Masking format for partial masking	Mandatory
Regular expression masking	regexp_pattern	varchar(1024)	Search pattern for regular expression masking	Mandatory
	regexp_replacement	varchar(1024)	Replacement character/string for regular expression masking	Mandatory
	regexp_flags	varchar(20)	Regular expression flags	NULL

- Change the masking policy description

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	action	varchar(63)	'SET_POLICY_DESCRIPTION'	Mandatory
	policy_description	varchar(1024)	Masking policy description	Mandatory

- Change the masking target description

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	action	varchar(63)	'SET_COLUMN_DESCRIPTION'	Mandatory
	column_name	varchar(63)	Masking target name	Mandatory
	column_description	varchar(1024)	Masking target description	Mandatory

Details about whether arguments can be omitted are as follows.

Argument	Mandatory or optional									
	ADD_COLUMN			DROP_COLUMN	MODIFY_EXPRESSION	MODIFY_COLUMN			SET_POLICY_DESCRIPTION	SET_COLUMN_DESCRIPTION
	Full masking	Partial masking	Regular expression masking			Full masking	Partial masking	Regular expression masking		
schema_name	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
table_name	N	N	N	N	N	N	N	N	N	N
policy_name	N	N	N	N	N	N	N	N	N	N
action	Y	Y	Y	N	N	N	N	N	N	N
column_name	N	N	N	N	-	N	N	N	-	N
function_type	Y	N	N	-	-	Y	N	N	-	-
expression	-	-	-	-	N	-	-	-	-	-
policy_description	-	-	-	-	-	-	-	-	N	-
column_description	-	-	-	-	-	-	-	-	-	N
function_parameters	-	N	-	-	-	-	N	-	-	-
regexp_pattern	-	-	N	-	-	-	-	N	-	-
regexp_replacement	-	-	N	-	-	-	-	N	-	-
regexp_flags	-	-	Y	-	-	-	-	Y	-	-

Y: Can be omitted; N: Cannot be omitted; -: Ignored when specified

Return value

Return value	Description
TRUE	Ended normally
FALSE	Ended abnormally

Execution example 1

Adding masking policy p1 to masking target c2

```
postgres=# select pgx_alter_confidential_policy(table_name := 't1', policy_name := 'p1', action :=
'ADD_COLUMN', column_name := 'c2', function_type := 'PARTIAL', function_parameters := 'VVVFVVVVFVVVV,
VVV-VVVV-VVVV, *, 4, 11');
 pgx_alter_confidential_policy
-------------------------------
```

```
 t
(1 row)
```

Execution example 2

Deleting masking target c1 from masking policy p1

```
postgres=# select pgx_alter_confidential_policy(table_name := 't1', policy_name := 'p1', action :=
'DROP_COLUMN', column_name := 'c1');
 pgx_alter_confidential_policy
-------------------------------
 t
(1 row)
```

Execution example 3

Changing the masking condition for masking policy p1

```
postgres=# select pgx_alter_confidential_policy(table_name := 't1', policy_name := 'p1', action :=
'MODIFY_EXPRESSION', expression := 'false');
 pgx_alter_confidential_policy
-------------------------------
 t
(1 row)
```

Execution example 4

Changing the content of masking policy p1 set for masking target c2

```
postgres=# select pgx_alter_confidential_policy(table_name := 't1', policy_name := 'p1', action :=
'MODIFY_COLUMN', column_name := 'c2', function_type := 'FULL');
 pgx_alter_confidential_policy
-------------------------------
 t
(1 row)
```

Execution example 5

Changing the description of masking policy p1

```
postgres=# select pgx_alter_confidential_policy(table_name := 't1', policy_name := 'p1', action :=
'SET_POLICY_DESCRIPTION', policy_description := 'this policy is an example.');
 pgx_alter_confidential_policy
-------------------------------
 t
(1 row)
```

Execution example 6

Changing the description of masking target c2

```
postgres=# select pgx_alter_confidential_policy(table_name := 't1', policy_name := 'p1', action :=
'SET_COLUMN_DESCRIPTION', column_name := 'c2', column_description := 'c2 column is FULL.');
 pgx_alter_confidential_policy
-------------------------------
 t
(1 row)
```

Description

- The arguments for the pgx_alter_confidential_policy system management function can be specified in any order.

- The action parameters below can be specified. When action parameters are omitted, ADD_COLUMN is applied.

Parameter	Description
ADD_COLUMN	Adds a masking target to a masking policy.
DROP_COLUMN	Deletes a masking target from a masking policy.
MODIFY_EXPRESSION	Changes expression.
MODIFY_COLUMN	Changes the content of a masking policy set for a masking target.
SET_POLICY_DESCRIPTION	Changes policy_description.
SET_COLUMN_DESCRIPTION	Changes column_description.

- The function_parameters argument is enabled when the function_type is PARTIAL. If the function_type is other than PARTIAL, it will be ignored.
- The arguments below are enabled when the function_type is REGEXP. If the function_type is other than REGEXP, these arguments will be ignored.
  - regexp_pattern
  - regexp_replacement
  - regexp_flags

- Refer to "String Constants" in the PostgreSQL Documentation for information on the strings to specify for arguments.
- Refer to "POSIX Regular Expressions" in the PostgreSQL Documentation and check pattern, replacement, and flags for information on the values that can be specified for regexp_pattern, regexp_replacement, and regexp_flags.

B.3.2 pgx_create_confidential_policy

Description

Creates masking policies

Format

The format varies depending on the masking type. The format is shown below.

```
pgx_create_confidential_policy(
[schema_name        := 'schemaName',]
table_name        := 'tableName',
policy_name         := 'policyName',
expression        := 'expr'
[, enable           := 'policyStatus']
[, policy_description    := 'policyDesc']
[, column_name        := 'colName'
     [, function_type    := 'FULL'] |
     [, function_type    := 'PARTIAL', partialOpt] |
     [, function_type    := 'REGEXP', regexpOpt]
     [, column_description    := 'colDesc']
])


partialOpt:
function_parameters    := 'maskingFmt'
```

```
regexpOpt:
regexp_pattern        := 'regexpPattern',
regexp_replacement        := 'regexpReplacementChar',
[, regexp_flags        := 'regexpFlags']
```

Argument

Details are as follows.

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	schema_name	varchar(63)	Schema name of table for which the masking policy is created	'public'
	table_name	varchar(63)	Name of table for which the masking policy is created	Mandatory
	policy_name	varchar(63)	Masking policy name	Mandatory
	expression	varchar(1024)	Masking condition	Mandatory
	enable	boolean	Masking policy status - 't': Enabled - 'f': Disabled	't'
	policy_description	varchar(1024)	Masking policy description	NULL
	column_name	varchar(63)	Masking target name	NULL
	function_type	varchar(63)	Masking type - 'FULL': Full masking - 'PARTIAL': Partial masking - 'REGEXP': Regular expression masking	'FULL'
	column_description	varchar(1024)	Masking target description	NULL
Partial masking	function_parameters	varchar(1024)	Masking format for partial masking	Mandatory
Regular expression masking	regexp_pattern	varchar(1024)	Search pattern for regular expression masking	Mandatory
	regexp_replacement	varchar(1024)	Replacement character/string for regular expression masking	Mandatory
	regexp_flags	varchar(20)	Regular expression flags	NULL

Details about whether arguments can be omitted are as follows.

Argument	Mandatory or optional		
	Full masking	Partial masking	Regular expression masking
schema_name	Y	Y	Y
table_name	N	N	N
policy_name	N	N	N
expression	N	N	N
enable	Y	Y	Y
policy_description	Y	Y	Y

Argument	Mandatory or optional		
	Full masking	Partial masking	Regular expression masking
column_name	Y	Y	Y
function_type	Y	Y	Y
column_description	Y	Y	Y
function_parameters	-	N	-
regexp_pattern	-	-	N
regexp_replacement	-	-	N
regexp_flags	-	-	Y

Y: Can be omitted; N: Cannot be omitted; -: Ignored when specified

Return value

Return value	Description
TRUE	Ended normally
FALSE	Ended abnormally

Execution example 1

Creating masking policy p1 that does not contain a masking target

```
postgres=# select pgx_create_confidential_policy(table_name := 't1', policy_name := 'p1',
expression := '1=1');
 pgx_create_confidential_policy
--------------------------------
 t
(1 row)
```

Execution example 2

Creating masking policy p1 that contains masking target c1 of which the masking type is full masking

```
postgres=# select pgx_create_confidential_policy(schema_name := 'public', table_name := 't1',
policy_name := 'p1', expression := '1=1', enable := 't', policy_description := 'this policy is an
example.', column_name := 'c1', function_type := 'FULL', column_description := 'c1 column is FULL.');
 pgx_create_confidential_policy
--------------------------------
 t
(1 row)
```

Execution example 3

Creating masking policy p1 that contains masking target c2 of which the masking type is partial masking

```
postgres=# select pgx_create_confidential_policy( table_name := 't1', policy_name := 'p1',
expression := '1=1', column_name := 'c2', function_type := 'PARTIAL', function_parameters :=
'VVVFVVVVFVVVV, VVV-VVVV-VVVV, *, 4, 11');
 pgx_create_confidential_policy
--------------------------------
 t
(1 row)
```

Execution example 4

Creating masking policy p1 that contains masking target c3 of which the masking type is regular expression masking

```
postgres=# select pgx_create_confidential_policy( table_name := 't1', policy_name := 'p1',
expression := '1=1', column_name := 'c3', function_type := 'REGEXP', regexp_pattern := '(.*)(@.*)',
regexp_replacement := 'xxx\2', regexp_flags := 'g');
 pgx_create_confidential_policy
--------------------------------
 t
(1 row)
```

Description

- The arguments for the pgx_create_confidential_policy system management function can be specified in any order.
- If column_name is omitted, only masking policies that do not contain masking target will be created.
- One masking policy can be created for each table. Use the pgx_alter_confidential_policy system management function to add a masking target to a masking policy.
- The function_parameters argument is enabled when the function_type is PARTIAL. If the function_type is other than PARTIAL, it will be ignored.
- The arguments below are enabled when the function_type is REGEXP. If the function_type is other than REGEXP, these arguments will be ignored.
  - regexp_pattern
  - regexp_replacement
  - regexp_flags

If a table for which a masking policy is to be applied is deleted, delete the masking policy as well.

- Refer to "String Constants" in the PostgreSQL Documentation for information on the strings to specify for arguments.
- Refer to "POSIX Regular Expressions" in the PostgreSQL Documentation and check pattern, replacement, and flags for information on the values that can be specified for regexp_pattern, regexp_replacement, and regexp_flags.

B.3.3 pgx_drop_confidential_policy

Description

Deletes masking policies

Format

```
pgx_drop_confidential_policy(
[schema_name    := 'schemaName', ]
table_name     := 'tableName',
policy_name    := 'policyName'
)
```

Argument

Details are as follows.

Argument	Data type	Description	Default value
schema_name	varchar(63)	Schema name of table for which a masking policy is deleted	'public'
table_name	varchar(63)	Name of table for which a masking policy is deleted	Mandatory
policy_name	varchar(63)	Masking policy name	Mandatory

Details about whether arguments can be omitted are as follows.

Argument	Mandatory or optional
schema_name	Y
table_name	N
policy_name	N

Y: Can be omitted; N: Cannot be omitted

Return value

Return value	Description
TRUE	Ended normally
FALSE	Ended abnormally

Execution example

Deleting masking policy p1

```
postgres=# select pgx_drop_confidential_policy(table_name := 't1', policy_name := 'p1');
 pgx_drop_confidential_policy
------------------------------
 t
(1 row)
```

Description

The arguments for the pgx_drop_confidential_policy system management function can be specified in any order.

Note

If a table for which a masking policy is to be applied is deleted, delete the masking policy as well.

See

Refer to "String Constants" in the PostgreSQL Documentation for information on the strings to specify for arguments.

B.3.4 pgx_enable_confidential_policy

Description

Enables or disables masking policies

Format

```
pgx_enable_confidential_policy(
[schema_name    := 'schemaName', ]
table_name     := 'tableName',
```

```
policy_name    := 'policyName',
enable         := 'policyStatus'
)
```

Argument

Details are as follows.

Argument	Data type	Description	Default value
schema_name	varchar(63)	Schema name of table for which a masking policy is enabled or disabled	'public'
table_name	varchar(63)	Name of table for which a masking policy is enabled or disabled	Mandatory
policy_name	varchar(63)	Masking policy name	Mandatory
enable	boolean	Masking policy status - 't': Enabled - 'f': Disabled	Mandatory

Details about whether arguments can be omitted are as follows.

Argument	Mandatory or optional
schema_name	Y
table_name	N
policy_name	N
enable	N

Y: Can be omitted; N: Cannot be omitted

Return value

Return value	Description
TRUE	Ended normally
FALSE	Ended abnormally

Execution example

Enabling masking policy p1

```
postgres=# select pgx_enable_confidential_policy(table_name := 't1', policy_name := 'p1', enable :=
't');
 pgx_enable_confidential_policy
--------------------------------
 t
(1 row)
```

Description

The arguments for the pgx_enable_confidential_policy system management function can be specified in any order.

Refer to "String Constants" in the PostgreSQL Documentation for information on the strings to specify for arguments.

B.3.5 pgx_update_confidential_values

Description

Changes replacement characters when full masking is specified for masking type

Format

```
pgx_update_confidential_values(
[number_value    := 'numberValue']
[, char_value    := 'charValue']
[, varchar_value    := 'varcharValue']
[, date_value    := 'dateValue']
[, ts_value    := 'tsValue']
)
```

Argument

Details are as follows.

Argument	Data type	Description
number_value	integer	Replacement character in numeric type
char_value	varchar(1)	Replacement character in char type
varchar_value	varchar(1)	Replacement character in varchar type
date_value	date	Replacement character in date type
ts_value	timestamp	Replacement character in timestamp type

Return value

Return value	Description
TRUE	Ended normally
FALSE	Ended abnormally

Execution example

Using '*' as a replacement character in char type and varchar type

```
postgres=# select pgx_update_confidential_values(char_value := '*', varchar_value := '*');
 pgx_update_confidential_values
--------------------------------
 t
(1 row)
```

Description

- The arguments for the pgx_update_confidential_values system management function can be specified in any order.
- Specify one or more arguments for the pgx_update_confidential_values system management function. A replacement character is not changed for an omitted argument.

See

Refer to "String Constants" in the PostgreSQL Documentation for information on the strings to specify for arguments.

B.4 VCI Data Load Control Function

The table below lists the function that loads VCI data to buffer cache.

Table B.4 VCI data load control function

Name	Return type	Description
pgx_prewarm_vci(vci_index regclass)	int8	Loads the VCI data to buffer cache.

pgx_prewarm_vci loads the specified VCI data to buffer cache and returns the number of blocks of the loaded VCI data.

The aggregation process using VCI may take time immediately after an instance is started, because the VCI data has not been loaded to buffer cache. Therefore, the first aggregation process can be sped up by executing pgx_prewarm_vci after an instance is started.

The amount of memory required for preloading is the number of blocks returned by pgx_prewarm_vci multiplied by the size of one block.

This function can only be executed if the user has reference privilege to the VCI index and execution privilege to the pg_prewarm function.

B.5 High-Speed Data Load Control Functions

The table below lists the functions that can be used for high-speed data load.

Table B.5 High-speed data load control functions

Name	Return type	Description
pgx_loader	bigint	Creates dynamic shared memory, starts parallel workers and loads data
pgx_loader_recovery	smallint	Resolves in-doubt transactions

The pgx_loader command executes the above functions internally.

Appendix C System Views

This appendix describes how to use the system views in Fujitsu Enterprise Postgres.

See

Refer to "System Views" under "Internals" in the PostgreSQL Documentation for information on other system views.

C.1 pgx_tablespaces

The pgx_tablespaces view provides information related to the encryption of tablespaces.

Table C.1 pgx_tablespaces view

Column	Type	References	Description
spctablespace	oid	pg_tablespace.oid	Tablespace OID
spcencalgo	text		Tablespace encryption algorithm

The spcencalgo string displays one of the following values:

- none: Tablespace is not encrypted
- AES128: AES with key length of 128 bits
- AES256: AES with key length of 256 bits

C.2 pgx_stat_lwlock

The pgx_stat_lwlock view displays statistics related to lightweight locks, with each type of content displayed on a separate line.

Table C.2 pgx_stat_lwlock view

Column	Type	Description
lwlock_name	name	Name of the lightweight lock
total_waits	bigint	Number of waits caused by the lightweight lock
total_wait_time	double precision	Number of milliseconds spent in waits caused by the lightweight lock
stats_reset	timestamp with timezone	Last time at which this statistics was reset

C.3 pgx_stat_latch

The pgx_stat_latch view displays statistics related to latches, with each type of wait information within Fujitsu Enterprise Postgres displayed on a separate line.

Table C.3 pgx_stat_latch view

Column	Type	Description
latch_name	name	Name of the latch
total_waits	bigint	Number of waits caused a wait
total_wait_time	double precision	Number of milliseconds spent in waits caused by the latch
stats_reset	timestamp with timezone	Last time at which this statistic was reset

C.4 pgx_stat_walwriter

The pgx_stat_walwriter view displays statistics related to WAL writing, in a single line.

Table C.4 pgx_stat_walwriter view

Column	Type	Description
dirty_writes	bigint	Number of times old WAL buffers were written to the disk because the WAL buffer was full when WAL records were added
writes	bigint	Number of WAL writes
write_blocks	bigint	Number of WAL write blocks
total_write_time	double precision	Number of milliseconds spent on WAL writing
stats_reset	timestamp with timezone	Last time at which this statistic was reset

C.5 pgx_stat_sql

The pgx_stat_sql view displays statistics related to SQL statement executions, with each type of SQL statement displayed on a separate line.

Table C.5 pgx_stat_sql view

Column	Type	Description
selects	bigint	Number of SELECT statements executed In database multiplexing mode, this number includes the SELECT statements executed in Mirroring Controller. Mirroring Controller executes the SELECT statement using the interval specified for the heartbeat_interval of the server definition file (milliseconds).
inserts	bigint	Number of INSERT statements executed
deletes	bigint	Number of DELETE statements executed
updates	bigint	Number of UPDATE statements executed
selects_with_parallelism	bigint	Number of times parallel scan was used in SELECT statements
inserts_with_parallelism	bigint	Not used
deletes_with_parallelism	bigint	Not used
updates_with_parallelism	bigint	Not used
copies_with_parallelism	bigint	Not used
declares	bigint	Number of DECLARE statements executed (number of cursor OPENs)
fetches	bigint	Number of FETCH statements executed
checkpoints	bigint	Number of CHECKPOINT statements executed
clusters	bigint	Number of CLUSTER statements executed
copies	bigint	Number of COPY statements executed
reindexes	bigint	Number of REINDEX statements executed
truncates	bigint	Number of TRUNCATE statements executed
locks	bigint	Number of times a lock occurred
stats_reset	timestamp with timezone	Last time at which this statistic was reset

C.6 pgx_stat_gmc

The pgx_stat_gmc view provides information about the GMC areas.

Table C.6 pgx_stat_gmc view

Column	Type	Description
searches	bigint	Number of times the cache table is searched.
hits	bigint	Number of times the cache table is hit.
size	bigint	The current amount of memory (bytes) used in the GMC area.
stats_reset	timestamp with timezone	Last time these statistics were reset.

C.7 pgx_stat_progress_loader

The pgx_stat_progress_loader view provides overall progress information for pgx_loader command.

The pgx_stat_progress_loader view displays the sum of the progress information of the back-end processes and the number of parallel worker processes when pgx_loader runs.

Table C.7 pgx_stat_progress_loader view

Column	Type	Description
pid	integer	Process ID of the backend.
datid	Oid	Oid of the database to connect to.
datname	name	Name of the database to connect to.
relid	Oid	Oid of the table to load.
command	text	Command executes the load process. (Always "COPY FROM"for pgx_loader)
type	text	Type of data source for the load operation.
bytes_processed	bigint	Size of the data at the end of the load. (Backend and worker totals)
bytes_total	bigint	Size of the data to load. (Backend and worker totals)
tuples_processed	bigint	Number of rows that have completed loading. (Backend and worker totals)
tuples_excluded	bigint	Number of rows skipped during the load process. (Backend and worker totals)

Appendix D Tables Used by Transparent Data Encryption

This appendix explains tables tables used by the transparent data encryption feature.

D.1 pgx_tde_master_key

Provides information about the master encryption key being used when using the key management system as a keystore.

Column	Type	Description
local_key_id	integer	Sequence of encryption keys used internally by Fujitsu Enterprise Postgres
kms_name	text	Key management system name
key_name	text	Not currently used
key_id	text	Key ID in key management system
start_time	timestamp	Time the key was activated
status	enum	in-use: Current master encryption key in use used: Master encryption key used in the past scheduled: A new encryption key that was requested to update the master encryption key and is to be used.

Execution example

```
postgres=# select * from pgx_tde_master_key;
local_key_id | kms_name  | key_name|              key_id                  |     start_time     | status
-------------+-----------+--------+--------------------------------------+--------------------+-----
           1  | mykmipsvr |        | a0eebc99-9c0b-0000-0000-000000000000 |2022-12-16 05:43:49| in-use
(1 row)
```

Note

Do not use queries with "*" in the selection list, as the order of the columns may change or columns may be added.

Appendix E Tables Used by Data Masking

This appendix explains tables used by the data masking feature.

Note

These tables are updated by the data masking control function, so do not use SQL statements to directly update these tables.

E.1 pgx_confidential_columns

This table provides information on masking target for which masking policies are set.

Column	Type	Description
schema_name	varchar(63)	Schema name of table for which a masking policy is applied
table_name	varchar(63)	Name of table for which a masking policy is applied
policy_name	varchar(63)	Masking policy name
column_name	varchar(63)	Masking target name
function_type	varchar(63)	Masking type - 'FULL': Full masking - 'PARTIAL': Partial masking - 'REGEXP': Regular expression masking
function_parameters	varchar(1024)	Masking format for partial masking
regexp_pattern	varchar(1024)	Search pattern for regular expression masking
regexp_replacement	varchar(1024)	Replacement character/string for regular expression masking
regexp_flags	varchar(20)	Regular expression flags
column_description	varchar(1024)	Masking target description

Execution example

```
postgres=# select * from pgx_confidential_columns;
 schema_name | table_name | policy_name | column_name | function_type |
function_parameters           | regexp_pattern | regexp_replacement | regexp_flags |
column_description
-------------+------------+-------------+-------------+---------------
+---------------------------------------+----------------+--------------------+-------------
+--------------------
 public      | t1         | p1          | c1          | FULL          |
|                |                     |             |
 public      | t1         | p1          | c2          | PARTIAL       | VVVFVVVVFVVVV, VVV-VVVV-VVVV,
*, 4, 11 |                |                     |             |
(2  row)
```

E.2 pgx_confidential_policies

This table provides information on masking policies.

Column	Type	Description
schema_name	varchar(63)	Schema name of table for which a masking policy is applied

Column	Type	Description
table_name	varchar(63)	Name of table for which a masking policy is applied
policy_name	varchar(63)	Masking policy name
expression	varchar(1024)	Masking condition
enable	boolean	Masking policy status - 't': Enabled - 'f': Disabled
policy_description	varchar(1024)	Masking policy description

Execution example

```
postgres=# select * from pgx_confidential_policies;
 schema_name | table_name | policy_name | expression | enable | policy_description
-------------+------------+-------------+------------+--------+--------------------
 public      | t1         | p1          | 1=1        | t      |
(1 row)
```

E.3 pgx_confidential_values

This table provides information on replacement characters when full masking is specified for masking type.

Column	Data type	Description	Default value
number_value	integer	Numeric	0
char_value	varchar(1)	char type	Spaces
varchar_value	varchar(1)	varchar type	Spaces
date_value	date	date type	'1970-01-01'
timestamp_value	timestamp	timestamp type	'1970-01-01 00:00:00'

Execution example

```
postgres=# select * from pgx_confidential_values;
 number_value | char_value | varchar_value | date_value |      ts_value
--------------+------------+---------------+------------+---------------------
            0 |            |               | 1970-01-01 | 1970-01-01 00:00:00
(1 row)
```

Appendix F Tables Used by High-Speed Data Load

This appendix describes the tables used by high-speed data load.

F.1 pgx_loader_state

The pgx_loader_state table provides information about transactions prepared by high-speed data load.

Column	Type	Description
id	serial	Unique identifier. This value is assigned from the pgx_loader_state_id_seq sequence.
gid	text	Global transaction identifier assigned to a transaction.
state	text	State of the transaction. The value can be one of the following: - commit: The prepared transaction has been committed. - rollback: The prepared transaction is in in-doubt state.
leader_pid	integer	Process ID of the backend process (leader process) that executed the pgx_loader control function.
role_oid	integer	Role identifier (OID). A prepared transaction can only be completed by the same user who executed the original transaction or by a superuser.
relation_oid	integer	Object identifier (OID).

Note

The pgx_loader_state table and pgx_loader_state_id_seq sequence are updated by high-speed data load. Do not update these database objects directly using SQL.

Appendix G Starting and Stopping the Web Server Feature of WebAdmin

To use WebAdmin for creating and managing a Fujitsu Enterprise Postgres instance on a server where Fujitsu Enterprise Postgres is installed, you must first start the Web server feature of WebAdmin.

- Using WebAdmin in a single-server configuration

  You must start the Web server on the server on which Fujitsu Enterprise Postgres and WebAdmin are installed.

- Using WebAdmin in a multiserver configuration

  You must start the Web server on all servers on which WebAdmin has been installed.

This appendix describes how to start and stop the Web server feature of WebAdmin.

Note that "<*x*>" in paths indicates the product version.

See

Refer to "Installing WebAdmin in a Multiserver Configuration" in the Installation and Setup Guide for Server for information on multiserver installation.

G.1 Starting the Web Server Feature of WebAdmin

Follow the procedure below to start the Web server feature of WebAdmin.

1. Change to superuser

   Acquire superuser privileges on the system.

   Example

   ```
   $ su -
   Password:******
   ```

2. Start the Web server feature of WebAdmin

   Execute the WebAdminStart command to start the Web server feature of WebAdmin.

   Example

   If WebAdmin is installed in /opt/fsepv<*x*>webadmin:

   ```
   # cd /opt/fsepv<x>webadmin/sbin
   # ./WebAdminStart
   ```

G.2 Stopping the Web Server Feature of WebAdmin

This section describes how to stop the Web server feature of WebAdmin.

Follow the procedure below to stop the Web server feature of WebAdmin.

1. Change to superuser

   Acquire superuser privileges on the system.

   Example

   ```
   $ su -
   Password:******
   ```

2. Stop the Web server feature of WebAdmin

   Execute the WebAdminStop command to stop the Web server feature of WebAdmin.

Example

If WebAdmin is installed in /opt/fsepv<*x*>webadmin:

```
# cd /opt/fsepv<x>webadmin/sbin
# ./WebAdminStop
```

Appendix H WebAdmin Wallet

This appendix describes how to use the Wallet feature of WebAdmin.

When a remote instance or a standby instance is created, it is necessary to provide user name and password for authentication with the remote machine or the database instance.

The Wallet feature in WebAdmin is a convenient way to create and store these credentials.

Once created, these credentials can be repeatedly used in one or more instances.

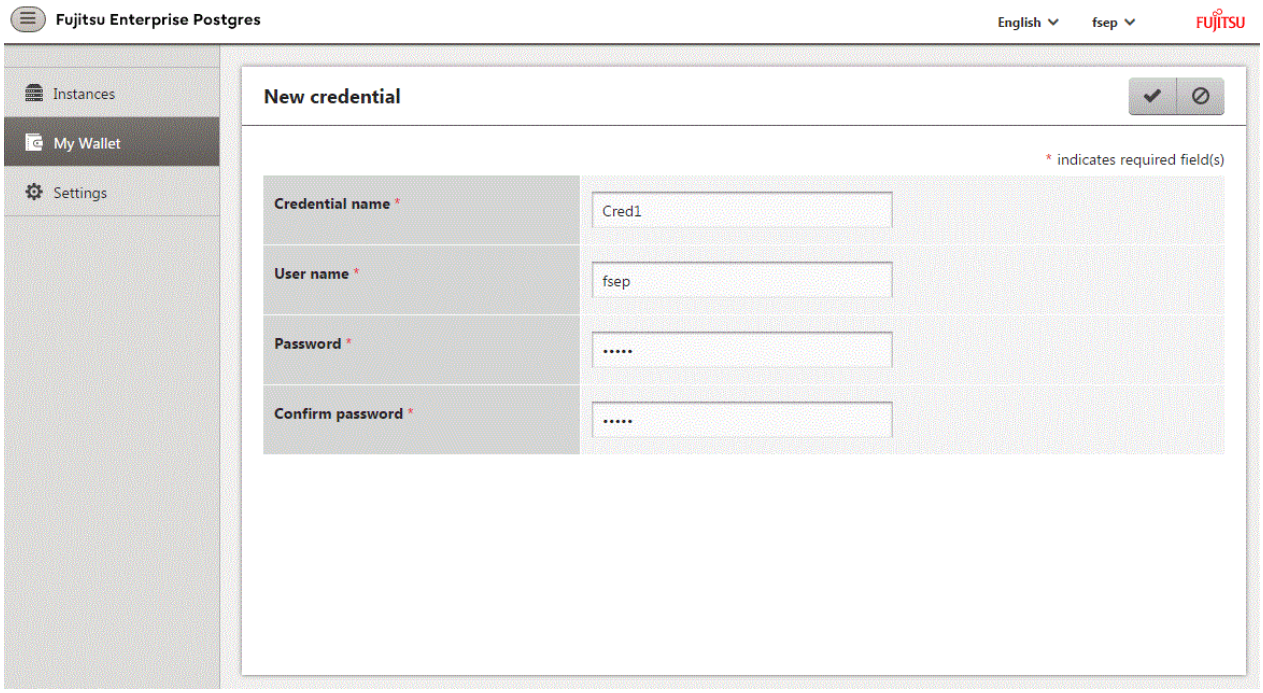
Note

It is not mandatory to create a credential in the Wallet. It is possible to create a remote instance or a standby instance without creating any credential in the Wallet.

If no credential is created beforehand, a user name and password can be entered in the instance creation page. When creating a "Remote" instance, if operating system credentials are entered without using a credential stored in the Wallet, WebAdmin automatically creates a credential with the given user name and password, and stores it in the user's wallet for future use.

H.1 Creating a Credential

1. In the [My Wallet] tab, click ➕. The [New credential] page will be displayed.
2. Enter the information for the credentials.



Enter the following items. Credential name, User name and Password should not contain hazardous characters. Refer to "Appendix I WebAdmin Disallow User Inputs Containing Hazardous Characters".

- [Credential name]: Name of the credential

  The name must meet the conditions below:

  - Maximum of 16 characters
  - The first character must be an ASCII alphabetic character
  - The other characters must be ASCII alphanumeric characters

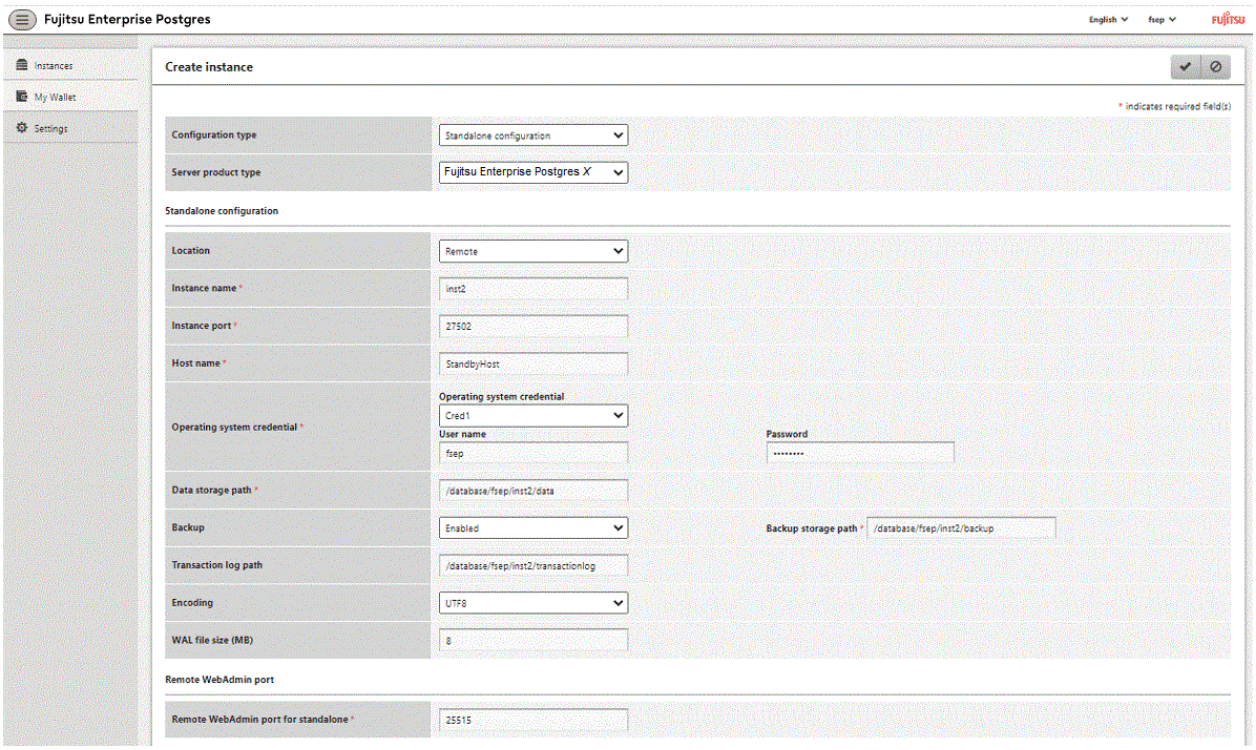
- [User name]: The operating system user name or database instance user name that will be used later
- [Password]: Password for the user
- [Confirm password]: Reenter the password.

3. Click ✔ to store the credential.

H.2 Using a Credential

Once a credential is created in the Wallet, it can be used during remote instance creation or standby instance creation.

The following page uses the credential that was created in the previous section.



When "Cred1" is selected in [Operating system credential], the user name and password are automatically populated from the credential.

Appendix I WebAdmin Disallow User Inputs Containing Hazardous Characters

WebAdmin considers the following as hazardous characters, which are not allowed in user inputs.

```
| (pipe sign)

& (ampersand sign)

; (semicolon sign)

$ (dollar sign)

% (percent sign)

@ (at sign)

' (single apostrophe)

" (quotation mark)

\' (backslash-escaped apostrophe)

\" (backslash-escaped quotation mark)

<> (triangular parenthesis)

() (parenthesis)

+ (plus sign)

CR (Carriage return, ASCII 0x0d)

LF (Line feed, ASCII 0x0a)

, (comma sign)

\ (backslash)
```

Appendix J Copy Command Samples that Use the Advanced Copy Feature of the ETERNUS Disk Array

Backup/recovery scripts that use OPC, an advanced copy feature of the Fujitsu Storage ETERNUS disk array (hereafter referred to as ETERNUS disk array), are supplied as copy command samples for use by the pgx_dmpall and pgx_rcvall commands.

Users can copy the samples to any file and make changes appropriate to their environment or operations.

The samples are stored in the directories below:

- Basic version

```
/installDir/share/copy_command.esf_acm1.sh.sample
```

- Advanced version

```
/installDir/share/copy_command.esf_acm2.sh.sample
```

See

The samples use the replication management command of the Fujitsu Storage ETERNUS SF AdvancedCopy Manager (hereafter referred to as ACM) for operating the advanced copy features of the ETERNUS disk array. Refer to the relevant manual for details.

Prerequisite for using the samples

To use the samples, it is necessary to configure the settings of the advanced copy features of the ETERNUS disk array in advance. The samples assume that the replication source/volume has been configured, its contents have been physically copied to the replication volume, and that the tracking process has been started.

Sample content (advanced version)

The sample uses two replication volumes as the replication source on which the database cluster and tablespace are located, alternating between them for each backup. It then registers to the backup information file the replication volume (group) used for the latest backup.

In addition, the backup information file is also used to determine the replication volume to restore during recovery.

The processing for each operation mode is described below:

prepare mode

1. The backup information file is read, and the replication volume to be used for the current backup is determined.
2. The completion status of physical copies to all replication volumes is checked using the swsrpstat command of ACM.
3. The replication volume determined in step 1 is written to a temporary file for later use by the backup mode.

backup mode

1. The temporary file is read, and the replication volume information is retrieved.
2. The file system buffer is written using the sync command of the operating system.
3. The file system targeted for backup is frozen using the fsfreeze command of the operating system.
4. Snapshot retrieval (implementation of a logical copy) is performed using the swsrpmake command of ACM.
5. The file system frozen in step 3 is unfrozen using the fsfreeze command of the operating system.

   Steps 3 to 5 are performed for all file systems targeted for backup.

finalize mode

1. The completion status of physical copies to all replication volumes is checked using the swsrpstat command of ACM.

2. Information about the replication volume used for the current backup is written to the backup information file.

restore mode

1. The backup information file is read, and the replication volume to be used for recovery is determined.
2. The file system targeted for recovery is unmounted using the umount command of the operating system.
3. Physical copy from the replication volume is performed using the swsrpmake command of ACM.
4. The file system targeted for recovery is mounted using the mount command of the operating system.
5. The completion status of physical copy from the replication volume is checked using the swsrpstat command of ACM.

   Steps 2 to 5 are performed for all file systems targeted for recovery.
6. Files and directories not needed for archive recovery are deleted.

Note

- The samples cannot be used on SLES 12 and SLES 15.
- The samples use the sudo command of the operating system so that operation of the commands and file systems of ACM is performed by the superuser of the operating system. Therefore, determine if these implementations satisfy the security standards on the database server, and if necessary, perform implementations using other means.
- The samples temporarily freeze the file system of the data storage destination (replication source volume) to protect the file system from copy processing by advanced copy features of the ETERNUS disk array. Therefore, consider the following:
  - Consider freeze time during the timeout period of an SQL statement.

    When using a feature that links with the database, such as a cluster feature that accesses the data storage destination, consider freeze time in relation to the timeout periods below.
    - When performing database multiplexing

      Misdetection may occur during abnormality monitoring, so it is necessary to consider the monitoring interval, timeout period and number of retries for abnormality monitoring, and consider temporarily stopping only the Mirroring Controller process during backup.
    - When performing failover operations using PRIMECLUSTER

      If a failure occurs while the file system is frozen, switching is triggered when PRIMECLUSTER detects an issue, but will be performed only after the system is unfrozen. In addition, if it takes time for the system to unfreeze, the active node may trigger operating system panic, resulting in a switch.

Appendix K Collecting Failure Investigation Data

If the cause of an error that occurs while building the environment or during operations is unclear, data must be collected for initial investigation.

This appendix describes how to collect data for initial investigation.

RHEL7,RHEL8,RHEL9 and SLES 12

Use FJQSS (Information Collection Tool) to collect data for initial investigation.

SLES 15

Use the pgx_fjqssinf command to collect data for initial investigation.

See

- Refer to the FJQSS manual for information on how to use FJQSS.
- Refer to the Reference for information on the pgx_fjqssinf command.

Note

- When using FJQSS to collect data for initial investigation, you must set the following environment variables:
  - Environment variables required for using Fujitsu Enterprise Postgres

    Refer to "Configure the environment variables" under the procedure for creating an instance in "Using the initdb Command" in the Installation and Setup Guide for Server for information on the values to be set in the environment variables.
  - PGDATA

    Set the data storage destination.
  - PGDATABASE

    Set the database name from which you want to collect data for initial investigation.
  - PGPORT

    Set the instance port number. This does not need to be set if the default port number (27500) has not been changed.
  - PGUSER

    Set the database superuser.
    Set the database superuser so that client authentication is possible.
    FJQSS establishes a TCP/IP connection with the template1 database and collects data from the database.
  - FSEP_HOME

    Set the Fujitsu Enterprise Postgres installation directory.
- Refer to "Collecting Failure Investigation Data" in the Cluster Operation Guide (Database Multiplexing) for information on how to collect failure investigation data when performing database multiplexing.

Index

[A]

Actions in Response to Instance Startup Failure....130
All user data within the specified tablespace....27
Approximate backup time....18
Approximate recovery time....103
Automatically opening the keystore....38

[B]

Backing Up and Recovering the Keystore....32
Backing Up and Restoring/Recovering the Database....34
Backup/Recovery Using the Copy Command....93
Backup and recovery using the pgx_dmpall and pgx_rcvall commands....34,44
backup cycle....19
Backup data....27
Backup operation....20
Backup operation (file backup)....21
Backup status....20,21
Backup using the backup information file....94
Backup Using the Copy Command....97
backup_destination (string)....139
Building and starting a standby server....38

[C]

Changing a Masking Policy....54
Changing the Keystore Passphrase....31
Changing the Master Encryption Key....31
Changing the master encryption key and the passphrase....38
Checking an Encrypted Tablespace....30,42
Checking backup status....98
Checking the operating status of an instance....14,16
Collecting Failure Investigation Data....176
Configuration of the Copy Command....93
Configuration of the copy command for backup....95
Configuration of the copy command for recovery....96
Confirming a Masking Policy....54
Continuous archiving and point-in-time recovery....35,45
Copy Command for Backup....99
Copy Command for Recovery....101
Copy Command Interface....99
Copy Command Samples that Use the Advanced Copy Feature of the ETERNUS Disk Array....174
core_contents (string)....137
core_directory (string)....137
Creating a Masking Policy....53
Cyclic usage of the backup area....94

[D]

Data Masking....48
Data Types for Masking....56
Deleting a Masking Policy....56
Determining the backup area of the latest backup....98

[E]

Enabling and Disabling a Masking Policy....55
Enabling Automatic Opening of the Keystore....31
Encrypting a Tablespace....29,42
Encrypting Existing Data....36,45
Encryption mechanisms....27,40
Errors in More Than One Storage Disk....130

[F]

Faster encryption and decryption based on hardware....27
File system level backup and restore....35,44

[H]

High-Speed Data Load....81

[I]

If failure occurred in the data storage disk or the transaction log storage disk....104
If failure occurred on the backup data storage disk....105,107
If failure occurred on the data storage disk or the transaction log storage directory....105
Importing and Exporting the Database....36,45
Installing and Operating the In-memory Feature....71

[K]

keystore_location (string)....137

[L]

Logging in to WebAdmin....2
log in....3

[M]

Managing the Keystore....31
Masking Condition....49
Masking Format....50
Masking Policy....48
Masking Target....49
Masking Type....49
Monitoring Database Activity....62

[O]

Opening the Keystore....29
Operating Fujitsu Enterprise Postgres....1

[P]

Parallel Query....79
Performing backup....97
Perform recovery....98
Periodic Backup....19
pgx_global_metacache (numerical value)....141
pgx_stat_gmc view....164
pgx_stat_latch view....162
pgx_stat_lwlock view....162
pgx_stat_progress_loader view....164
pgx_stat_sql view....163
pgx_stat_walwriter view....163
pgx_tablespaces....162
pgx_tablespaces view....162
Placement and automatic opening of the keystore file....37
Placing the keystore file....37

Preparing for backup..... 97

[R]

Recovery Using the Copy Command..... 98
reserve_buffer_ratio (numerical value)..... 140

[S]

Scope of encryption..... 27
search_path (string)..... 139
Security-Related Notes..... 38,46
Security Notes..... 57
Setting a restore point..... 22
Setting the Master Encryption Key..... 28,41
Starting and Stopping the Web Server Feature of WebAdmin 169
Starting an instance..... 13,15
Starting pgAdmin..... 3
Startup URL for WebAdmin..... 3
Stopping an instance..... 13,15
Streaming replication support..... 28
Streaming Replication Using WebAdmin..... 66
Strong encryption algorithms..... 27
System Administration Functions..... 146
System Views..... 162

[T]

tablespace_encryption_algorithm (string)..... 139
Tables Used by Data Masking 166
Tables Used by Transparent Data Encryption..... 165
tde_kms.enable_shared_dek (boolean)..... 137
tde_kms.kms_conninfo_file (string)..... 137
tde_kms.max_shared_dek (numerical value)..... 137
tde_kms.plugin_path (string)..... 137
Tips for Installing Built Applications..... 39,47
track_gmc (string)..... 142
track_sql (string)..... 140
track_waits (string)..... 140
Transparent Data Encryption Control Functions..... 146
Two-layer encryption key and the keystore..... 27,40

[U]

User environment..... 3
Using Server Commands..... 15

[V]

vci.control_max_workers (numerical value)..... 140
vci.cost_threshold (numeric)..... 140
vci.enable (string)..... 141
vci.log_query (string)..... 141
vci.maintenance_work_mem (numerical value)..... 141
vci.max_local_ros (numerical value)..... 141
vci.max_parallel_degree (numerical value)..... 141
vci.shared_work_mem (numerical value)..... 141

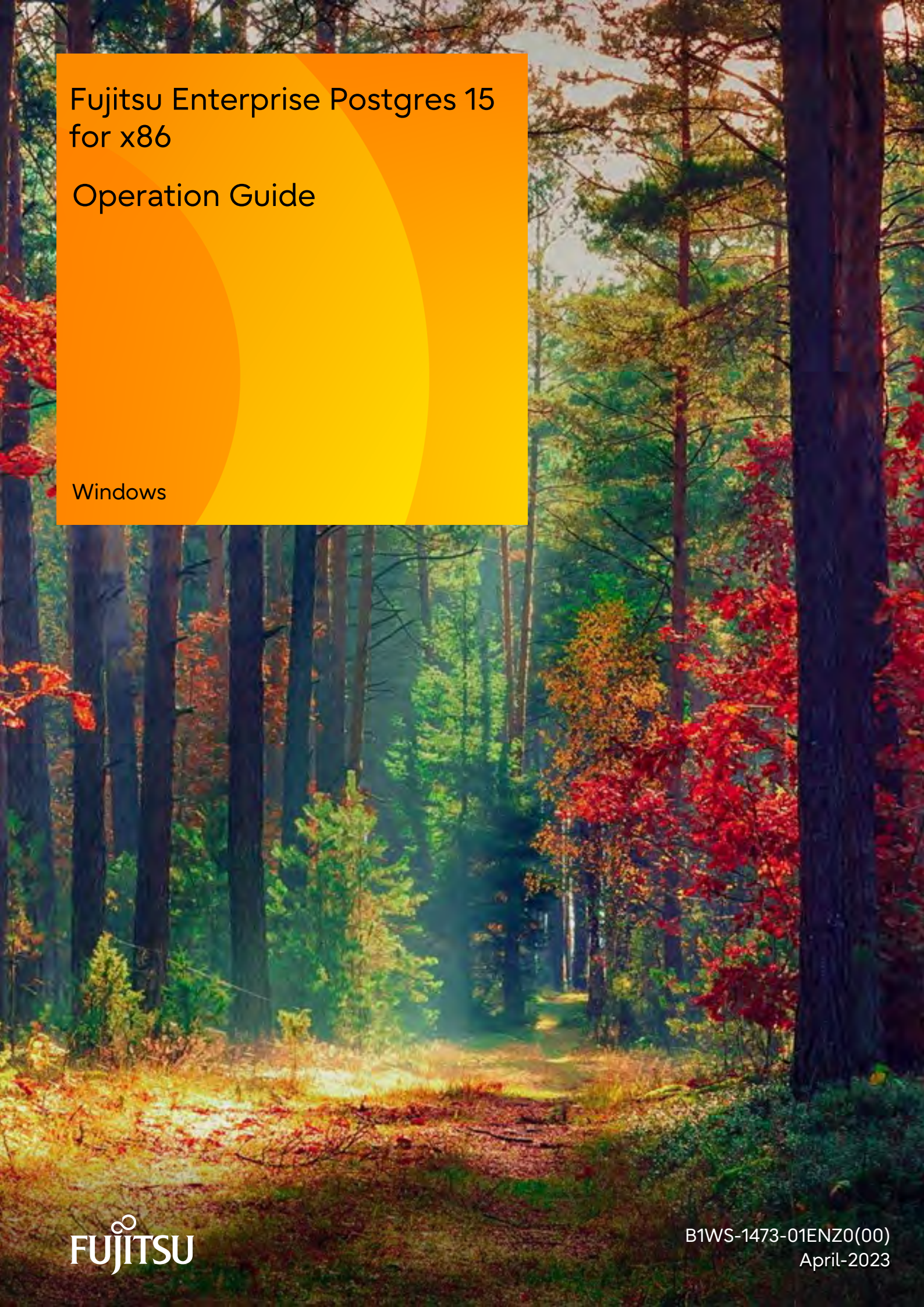
[W]

WAL and temporary files..... 28
WAL Mirroring Control Functions..... 146
WebAdmin Wallet..... 171

Fujitsu Enterprise Postgres 15 for x86

Operation Guide

Windows



FUJITSU

B1WS-1473-01ENZ0(00)
April-2023

Preface

Purpose of this document

The Fujitsu Enterprise Postgres database system extends the PostgreSQL features and runs on the Windows platform.

This document is the Fujitsu Enterprise Postgres Operation Guide.

Intended readers

This document is intended for those who install and operate Fujitsu Enterprise Postgres.

Readers of this document are assumed to have general knowledge of:

- PostgreSQL
- SQL
- Windows

Structure of this document

This document is structured as follows:

Chapter 1 Operating Fujitsu Enterprise Postgres

Describes how to operate Fujitsu Enterprise Postgres.

Chapter 2 Starting an Instance and Creating a Database

Describes how to start a Fujitsu Enterprise Postgres instance, and how to create a database.

Chapter 3 Backing Up the Database

Describes how to back up the database.

Chapter 4 Configuring Secure Communication Using Secure Sockets Layer

Describes communication data encryption between the client and the server.

Chapter 5 Protecting Storage Data Using Transparent Data Encryption

Describes how to encrypt the data to be stored in the database.

Chapter 6 Data Masking

Describes the data masking feature.

Chapter 7 Periodic Operations

Describes the periodic database operations that must be performed on Fujitsu Enterprise Postgres.

Chapter 8 Streaming Replication Using WebAdmin

Describes how to create a streaming replication cluster using WebAdmin.

Chapter 9 Installing and Operating the In-memory Feature

Describes how to install and operate the in-memory feature.

Chapter 10 Parallel Query

Describes the factors taken into consideration by Fujitsu Enterprise Postgres when performing parallel queries.

Chapter 11 High-Speed Data Load

Describes how to install and operate high-speed data load.

Chapter 12 Global Meta Cache

Describes how to use Grobal Meta Cache feature.

Chapter 13 Local Meta Cache Limit

Describes how to use Local Meta Cache Limit feature.

Chapter 14 Backup/Recovery Using the Copy Command

Describes backup and recovery using the copy command created by the user.

Chapter 15 Actions when an Error Occurs

Describes how to perform recovery when disk failure or data corruption occurs.

Appendix A Parameters

Describes the Fujitsu Enterprise Postgres parameters.

Appendix B System Administration Functions

Describes the system administration functions of Fujitsu Enterprise Postgres.

Appendix C System Views

Describes how to use the system view in Fujitsu Enterprise Postgres.

Appendix D Tables Used by Data Masking

Describes the tables used by the data masking feature.

Appendix E Tables Used by High-Speed Data Load

Describes the tables used by high-speed data load.

Appendix F Starting and Stopping the Web Server Feature of WebAdmin

Describes how to start and stop WebAdmin (Web server feature).

Appendix G WebAdmin Wallet

Describes how to use the Wallet feature of WebAdmin.

Appendix H WebAdmin Disallow User Inputs Containing Hazardous Characters

Describes characters not allowed in WebAdmin.

Appendix I Collecting Failure Investigation Data

Describes how to collect information for initial investigation.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

```
Edition 1.0: April  2023
```

Copyright

Copyright 2015-2023 Fujitsu Limited

Contents

Chapter 1 Operating Fujitsu Enterprise Postgres....1
1.1 Operating Methods....1
1.2 Starting WebAdmin....2
1.2.1 Logging in to WebAdmin....3
1.3 Starting pgAdmin....3
1.3.1 Starting pgAdmin....4
1.3.2 Adding an Instance....4
1.3.3 Connecting/Disconnecting an Instance....6
1.3.4 Changing the pgAdmin Display Language....6
1.4 Operations Using Commands....6
1.5 Operating Environment of Fujitsu Enterprise Postgres....7
1.5.1 Operating Environment....7
1.5.2 File Composition....9
1.6 Notes on Compatibility of Applications Used for Operations....10
1.7 Notes on Upgrading Database Instances....10
1.7.1 Additional Steps for upgrading to Fujitsu Enterprise Postgres with Vertical Clustered Index (VCI) Enabled....11

Chapter 2 Starting an Instance and Creating a Database....13
2.1 Starting and Stopping an Instance....13
2.1.1 Using WebAdmin....13
2.1.2 Using Commands....15
2.2 Creating a Database....16
2.2.1 Using pgAdmin....16
2.2.2 Using Client Commands....17

Chapter 3 Backing Up the Database....18
3.1 Periodic Backup....19
3.2 Backup Methods....19
3.2.1 Using WebAdmin....19
3.2.2 Using Server Commands....20

Chapter 4 Configuring Secure Communication Using Secure Sockets Layer....24
4.1 Configuring Communication Data Encryption....24
4.1.1 Issuing a Certificate....25
4.1.2 Deploying a Server Certificate File and a Server Private Key File....25
4.1.3 Distributing a CA Certificate File to the Client....25
4.1.4 Configuring the Operating Environment for the Database Server....25
4.1.5 Configuring the Operating Environment for the Client....25
4.1.6 Performing Database Multiplexing....26

Chapter 5 Protecting Storage Data Using Transparent Data Encryption....27
5.1 Protecting Data Using Encryption....27
5.2 Setting the Master Encryption Key....28
5.3 Opening the Keystore....28
5.4 Encrypting a Tablespace....29
5.5 Checking an Encrypted Tablespace....30
5.6 Managing the Keystore....31
5.6.1 Changing the Master Encryption Key....31
5.6.2 Changing the Keystore Passphrase....31
5.6.3 Enabling Automatic Opening of the Keystore....31
5.6.4 Backing Up and Recovering the Keystore....32
5.7 Backing Up and Restoring/Recovering the Database....34
5.8 Importing and Exporting the Database....36
5.9 Encrypting Existing Data....36
5.10 Operations in Cluster Systems....37
5.10.1 HA Clusters that do not Use Database Multiplexing....37

5.10.2 Database Multiplexing Mode....37
5.11 Security-Related Notes....38
5.12 Tips for Installing Built Applications....39

Chapter 6 Data Masking....40
6.1 Masking Policy....40
6.1.1 Masking Target....41
6.1.2 Masking Type....41
6.1.3 Masking Condition....41
6.1.4 Masking Format....42
6.2 Usage Method....44
6.2.1 Creating a Masking Policy....45
6.2.2 Changing a Masking Policy....46
6.2.3 Confirming a Masking Policy....46
6.2.4 Enabling and Disabling a Masking Policy....47
6.2.5 Deleting a Masking Policy....48
6.3 Data Types for Masking....48
6.4 Security Notes....49

Chapter 7 Periodic Operations....50
7.1 Configuring and Monitoring the Log....50
7.2 Monitoring Disk Usage and Securing Free Space....50
7.2.1 Monitoring Disk Usage....50
7.2.2 Securing Free Disk Space....50
7.3 Automatically Closing Connections....51
7.4 Monitoring the Connection State of an Application....51
7.4.1 Using the View (pg_stat_activity)....52
7.4.2 Using pgAdmin....52
7.5 Reorganizing Indexes....53
7.6 Monitoring Database Activity....54
7.6.1 Information that can be Collected....55
7.6.2 Collection Configuration....56
7.6.3 Information Reset....56

Chapter 8 Streaming Replication Using WebAdmin....58
8.1 Creating a Standby Instance....58
8.2 Promoting a Standby Instance....60
8.3 Converting an Asynchronous Replication to Synchronous....60
8.4 Converting a Synchronous Replication to Asynchronous....61
8.5 Joining a Replication Cluster....61

Chapter 9 Installing and Operating the In-memory Feature....63
9.1 Installing Vertical Clustered Index (VCI)....63
9.1.1 Evaluating whether to Install VCI....63
9.1.2 Estimating Resources....64
9.1.3 Setting up....64
9.1.3.1 Setting Parameters....64
9.1.3.2 Installing the Extensions....65
9.1.3.3 Creating a VCI....66
9.1.3.4 Confirming that the VCI has been Created....66
9.1.4 Data that can Use VCI....66
9.1.4.1 Relation Types....66
9.1.4.2 Data Types....67
9.2 Operating VCI....69
9.2.1 Commands that cannot be Used for VCI....69
9.2.2 Data Preload Feature....70

Chapter 10 Parallel Query....72

10.1 CPU Load Calculation....72
10.2 Increase of Workers during Runtime....72
10.3 Statistics View Displays the Action State....72

Chapter 11 High-Speed Data Load....74
11.1 Installing High-Speed Data Load....74
11.1.1 Deciding whether to Install....74
11.1.2 Estimating Resources....74
11.1.3 Setup....75
11.1.3.1 Setting Parameters....75
11.1.3.2 Installing the Extension....76
11.2 Using High-Speed Data Load....76
11.2.1 Loading Data....76
11.2.2 Checking Progress....77
11.2.3 Recovering from a Data Load that Ended Abnormally....78
11.3 Removing High-Speed Data Load....79
11.3.1 Removing the Extension....79

Chapter 12 Global Meta Cache....81
12.1 Usage....81
12.1.1 Deciding Whether to Enable the Global Meta Cache Feature....81
12.1.2 Estimating Memory for Global Meta Cache....81
12.1.3 How the GMC Memory Area Is Used....81
12.1.4 Enabling the Global Meta Cache Feature....81
12.1.5 Estimating Resources....82
12.2 Statistics....82
12.2.1 System View....82

Chapter 13 Local Meta Cache Limit....83
13.1 Usage....83
13.1.1 Deciding Whether to Enable the Local Meta Cache Limit Feature....83
13.1.2 How to Set Parameters for the Local Meta Cache Limit Feature....83
13.1.3 Cache Removal when Local Meta Cache Limit is Enabled....83
13.1.4 Performance Impact and Parameter Tuning of the Local Meta Cache Limit Feature....84

Chapter 14 Backup/Recovery Using the Copy Command....86
14.1 Configuration of the Copy Command....86
14.2 Backup Using the Copy Command....89
14.3 Recovery Using the Copy Command....90
14.4 Copy Command Interface....91
14.4.1 Copy Command for Backup....91
14.4.2 Copy Command for Recovery....93

Chapter 15 Actions when an Error Occurs....95
15.1 Recovering from Disk Failure (Hardware)....96
15.1.1 Using WebAdmin....96
15.1.2 Using Server Command....97
15.2 Recovering from Data Corruption....101
15.2.1 Using WebAdmin....101
15.2.2 Using the pgx_rcvall Command....102
15.3 Recovering from an Incorrect User Operation....103
15.3.1 Using WebAdmin....103
15.3.2 Using the pgx_rcvall Command....104
15.4 Actions in Response to an Application Error....105
15.4.1 When using the view (pg_stat_activity)....105
15.4.2 Using pgAdmin....106
15.5 Actions in Response to an Access Error....106
15.6 Actions in Response to Insufficient Space on the Data Storage Destination....107

15.6.1 Using a Tablespace ... 107
15.6.2 Replacing the Disk with a Larger Capacity Disk ... 107
15.6.2.1 Using WebAdmin ... 107
15.6.2.2 Using Server Commands ... 108
15.7 Actions in Response to Insufficient Space on the Backup Data Storage Destination ... 109
15.7.1 Temporarily Saving Backup Data ... 109
15.7.1.1 Using WebAdmin ... 110
15.7.1.2 Using Server Commands ... 110
15.7.2 Replacing the Disk with a Larger Capacity Disk ... 113
15.7.2.1 Using WebAdmin ... 113
15.7.2.2 Using Server Commands ... 114
15.8 Actions in Response to Insufficient Space on the Transaction Log Storage Destination ... 118
15.8.1 Replacing the Disk with a Larger Capacity Disk ... 118
15.8.1.1 Using WebAdmin ... 118
15.8.1.2 Using Server Commands ... 119
15.9 Errors in More Than One Storage Disk ... 120
15.10 Actions in Response to Instance Startup Failure ... 121
15.10.1 Errors in the Configuration File ... 121
15.10.2 Errors Caused by Power Failure or Mounting Issues ... 121
15.10.3 Other Errors ... 121
15.10.3.1 Using WebAdmin ... 122
15.10.3.2 Using Server Commands ... 122
15.11 Actions in Response to Failure to Stop an Instance ... 122
15.11.1 Using WebAdmin ... 122
15.11.2 Using Server Commands ... 122
15.11.2.1 Stopping the Instance Using the Fast Mode ... 123
15.11.2.2 Stopping the Instance Using the Immediate Mode ... 123
15.11.2.3 Forcibly Stopping the Server Process ... 123
15.12 Actions in Response to Failure to Create a Streaming Replication Standby Instance ... 123
15.13 Actions in Response to Error in a Distributed Transaction ... 124
15.14 I/O Errors Other than Disk Failure ... 125
15.14.1 Network Error with an External Disk ... 125
15.14.2 Errors Caused by Power Failure or Mounting Issues ... 125
15.15 Anomaly Detection and Resolution ... 125
15.15.1 Port Number and Backup Storage Path Anomalies ... 126
15.15.2 Mirroring Controller Anomalies ... 127

Appendix A Parameters ... 128

Appendix B System Administration Functions ... 135
B.1 WAL Mirroring Control Functions ... 135
B.2 Transparent Data Encryption Control Functions ... 135
B.2.1 pgx_open_keystore ... 135
B.2.2 pgx_set_master_key ... 135
B.2.3 pgx_set_keystore_passphrase ... 136
B.3 Data Masking Control Functions ... 136
B.3.1 pgx_alter_confidential_policy ... 136
B.3.2 pgx_create_confidential_policy ... 142
B.3.3 pgx_drop_confidential_policy ... 145
B.3.4 pgx_enable_confidential_policy ... 146
B.3.5 pgx_update_confidential_values ... 147
B.4 VCI Data Load Control Function ... 148
B.5 High-Speed Data Load Control Functions ... 149

Appendix C System Views ... 150
C.1 pgx_tablespaces ... 150
C.2 pgx_stat_lwlock ... 150
C.3 pgx_stat_latch ... 150

C.4 pgx_stat_walwriter....151
C.5 pgx_stat_sql....151
C.6 pgx_stat_gmc....152
C.7 pgx_stat_progress_loader....152

Appendix D Tables Used by Data Masking....153
D.1 pgx_confidential_columns....153
D.2 pgx_confidential_policies....153
D.3 pgx_confidential_values....154

Appendix E Tables Used by High-Speed Data Load....155
E.1 pgx_loader_state....155

Appendix F Starting and Stopping the Web Server Feature of WebAdmin....156
F.1 Starting the Web Server Feature of WebAdmin....156
F.2 Stopping the Web Server Feature of WebAdmin....156

Appendix G WebAdmin Wallet....157
G.1 Creating a Credential....157
G.2 Using a Credential....158

Appendix H WebAdmin Disallow User Inputs Containing Hazardous Characters....159

Appendix I Collecting Failure Investigation Data....160

Index....161

Chapter 1 Operating Fujitsu Enterprise Postgres

This chapter describes how to operate Fujitsu Enterprise Postgres.

1.1 Operating Methods

There are two methods of managing Fujitsu Enterprise Postgres operations:

- Operation management using GUI tools
- Operation management using commands

See

Before performing database multiplexing using database multiplexing, refer to "Database Multiplexing Mode" in the Cluster Operation Guide (Database Multiplexing).

Operation management using GUI tools

This involves managing operations using the WebAdmin and pgAdmin GUI tools.

- Management using WebAdmin

  This removes the requirement for complex environment settings and operational design for backup and recovery that is usually required for running a database. It enables you to easily and reliably monitor the state of the database, create a streaming replication cluster, back up the database, and restore it even if you do not have expert knowledge of databases.

- Management using pgAdmin

  When developing applications and maintaining the database, you can use pgAdmin to perform simple operations on database objects, such as:

  - Rebuild indexes and update statistics
  - Create, delete, and update database objects

  In addition, from pgAdmin of Fujitsu Enterprise Postgres, you can use the expanded features provided by Fujitsu Enterprise Postgres on the PostgreSQL SQL commands.

  See

  Refer to pgAdmin Help for information on the expanded features of pgAdmin provided by Fujitsu Enterprise Postgres.

Operation management using commands

You can use commands for configuring and operating the database and managing operations.

Note

- You cannot combine WebAdmin and server commands to perform the following operations:
  - Use commands to operate an instance created using WebAdmin.
  - Use WebAdmin to recover a database backed up using commands.

  For instances created with WebAdmin, however, backup can be obtained with the pgx_dmpall command. Also, WebAdmin can perform recovery by using the backup obtained with the pgx_dmpall command.

- To operate an instance created using the initdb command in WebAdmin, the instance needs to be imported using WebAdmin.

- You can perform backup and restoration in pgAdmin, but the backup data obtained with WebAdmin and pgx_dmpall is not compatible with the backup data obtained with pgAdmin.
- Refer to pgAdmin Help for other notes on pgAdmin.

Features used in each phase

The following table lists the features used in each phase for GUI-based operations and command-based operations.

Operation		Operation with the GUI	Operation with commands
Setup	Creating an instance	WebAdmin is used. The server machine capacity, and the optimum parameter for operations using WebAdmin, are set automatically.	The configuration file is edited directly using the initdb command.
	Creating a standby instance	WebAdmin is used. WebAdmin performs a base backup of the source instance and creates a standby instance.	A standby instance is created using the pg_basebackup command.
	Changing the configuration files	WebAdmin is used.	The configuration file is edited directly.
Starting and stopping an instance		WebAdmin is used.	The net command or sc command of the operating system is used.
Creating a database		This is defined using pgAdmin of the GUI tool, or using the psql command or the application after specifying the DDL statement.	
Backing up the database		WebAdmin, or the pgx_dmpall command, is used.	It is recommended that the pgx_dmpall command be used. Recovery to the latest database can be performed.
Database recovery		WebAdmin is used.	To use the backup that was performed using the pgx_dmpall command, the pgx_rcvall command is used.
Monitoring	Database errors	The status in the WebAdmin window can be checked. (*1)	The messages that are output to the database server log are monitored (*1)
	Disk space	The status in the WebAdmin window can be checked. A warning will be displayed if the free space falls below 20%. (*1)	This is monitored using the fsutil command (check free space), and the dir command (check used space), of the operating system, for example. (*1)
	Connection status	This can be checked using pgAdmin of the GUI tool, or referencing pg_stat_activity of the standard statistics view from psql or the application.	

*1: This can be used together with system log monitoring using operations management middleware (Systemwalker Centric Manager, for example).

1.2 Starting WebAdmin

This section describes how to start and log in to WebAdmin.

1.2.1 Logging in to WebAdmin

This section describes how to log in to WebAdmin.

User environment

It is recommended to use the following browsers with WebAdmin:

- Microsoft Edge (Build41 or later)

WebAdmin will work with other browsers, such as Firefox and Chrome, however, the look and feel may be slightly different.

Startup URL for WebAdmin

In the browser address bar, type the startup URL of the WebAdmin window in the following format:

```
http://hostNameOrIpAddress:portNumber/
```

- *hostNameOrIpAddress*: The host name or IP address of the server where WebAdmin is installed.
- *portNumber*: The port number of WebAdmin. The default port number is 27515.

Example

For a server with IP address "192.0.2.0" and port number "27515"

```
http://192.0.2.0:27515/
```

Display the startup windows. From this window you can log in to WebAdmin or access the product documentation.

Note

- You must start the Web server feature of WebAdmin before using WebAdmin.
- Refer to "Appendix F Starting and Stopping the Web Server Feature of WebAdmin" for information on how to start the Web server feature of WebAdmin.

Log in to WebAdmin

Click [Launch WebAdmin] in the startup URL window to start WebAdmin and display the login window.

To log in, specify the following values:

- [User name]: User name (OS user account) of the instance administrator
- [Password]: Password corresponding to the user name

Point

Use the OS user account as the user name of the instance administrator. Refer to "Creating an Instance Administrator" in the Installation and Setup Guide for Server for details.

1.3 Starting pgAdmin

This section describes how to start pgAdmin, how to add an instance required for managing a database, and how to connect to and disconnect from the instance.

You can use pgAdmin on the Windows client.

1.3.1 Starting pgAdmin

This section explains how to start pgAdmin if you are using it from the product "Fujitsu Enterprise Postgres Client (64bit) *x* SP*z*" (where "*x*" is the product version, and "*z*" is the product level (*x* SPz)).

Click [Start] >> [All apps] >> [Fujitsu Enterprise Postgres Client(64bit) *x* SP*z*] and start [pgAdmin 4 (64bit) (*x* SP*z*)].

Note

- You must start the instance to be connected to before using pgAdmin.
- Refer to "2.1 Starting and Stopping an Instance" for information on how to start an instance.
- When using pgAdmin4 with Microsoft Edge, enable network access by loopback in Microsoft Edge. Also, add Microsoft Edge to the loopback exclusion list.

1.3.2 Adding an Instance

This section describes how to add an instance to be connected to.

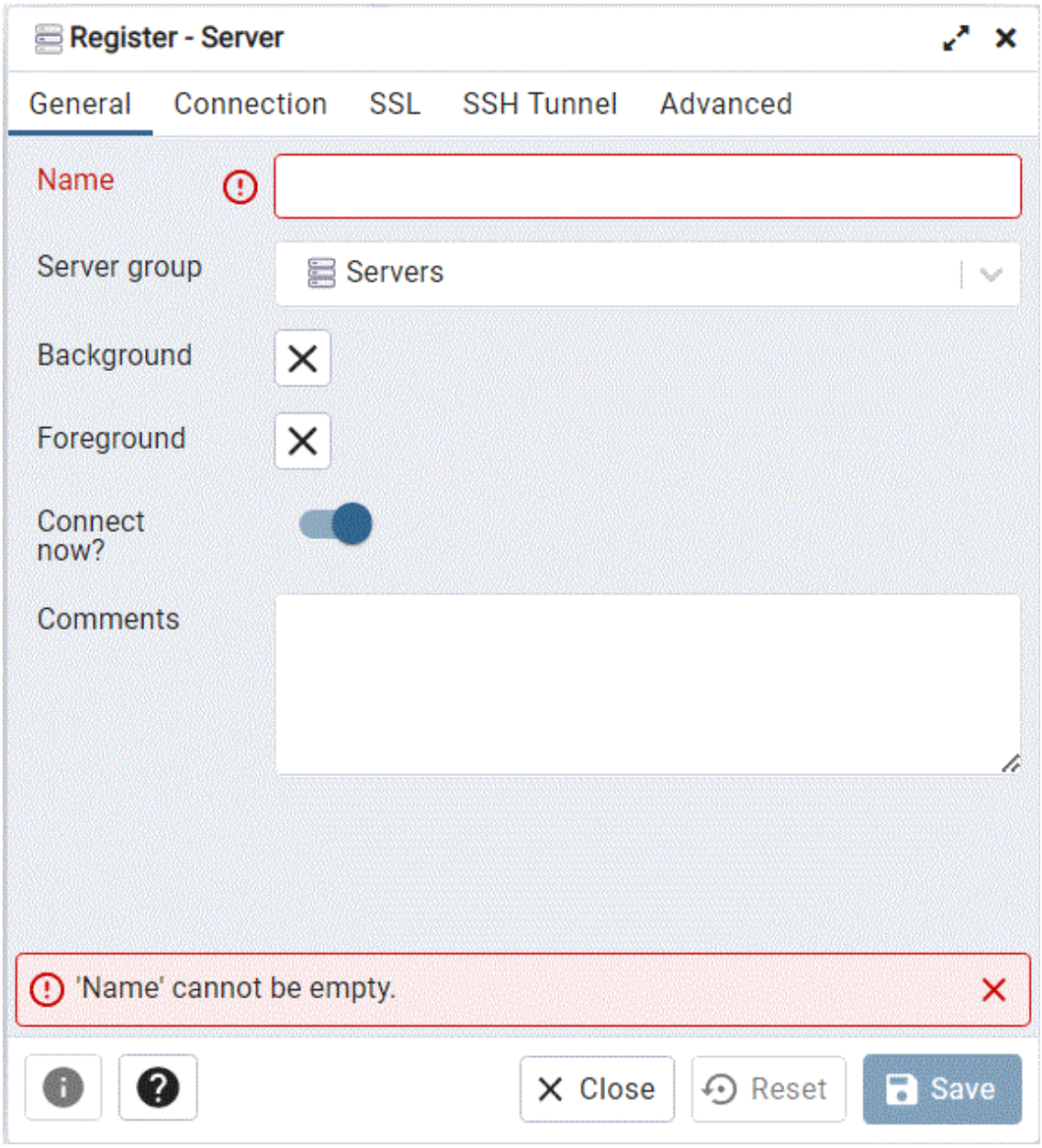
Note

If you use a link-local address with version 6 of the TCP/IP protocol, you may encounter the following error when registering the server with pgAdmin. Therefore, do not use link-local addresses.

```
unsupported format character '"' (0x22) at index 96
```

1. In the [Browser] pane, right-click [Servers], and then click [Register] >> [Server...].

2. In the [Register - Server] window, specify a value for each item.



[General] tab

- [Name]: Name of the instance to be managed

[Connection] tab

- [Host name/address]: Host name or IP address of the server where Fujitsu Enterprise Postgres is installed
- [Port]: Port number of the instance
- [Maintenance database]: Database name to which the client will connect
- [Username]: User name of the instance administrator
- [Password]: Password for the user name specified in [Username]

Note

If you select [Save password], the Fujitsu Enterprise Postgres connection password is stored in the following location. Set the appropriate access permissions for the password file to protect it from unauthorized access.

- %APPDATA%\Roaming\pgAdmin\pgadmin4.db

1.3.3 Connecting/Disconnecting an Instance

This section describes how to connect pgAdmin to an instance, and how to disconnect it.

Note

To connect to an instance created using WebAdmin, you must first configure the settings in the [Client authentication] window of WebAdmin to permit connection from pgAdmin.

See

Refer to "Changing the settings" in the Installation and Setup Guide for Server for information on the [Client authentication] window of WebAdmin.

Connecting to an instance

Starting pgAdmin does not connect it to any instance.

To connect to an instance, in the [Browser] pane, right-click the server, and then click [Connect Server].

If a password was not saved when the instance was added, enter a password in the password entry window that is displayed.

Disconnecting from an instance

To disconnect from an instance, in the [Browser] pane, right-click the server, and then click [Disconnect from Server].

1.3.4 Changing the pgAdmin Display Language

This section describes the procedure for changing the display language of pgAdmin.

1. Start pgAdmin.
2. Click [File] >> [Preferences].
3. Click [Miscellaneous] >> [User language] in the left pane of the [Preferences] dialog.
4. Select the language you want to display from the [User language] drop-down list displayed in the right pane, and click the [Save] button.
5. Refresh page by clicking the [Refresh] button.

1.4 Operations Using Commands

You can operate and manage the database using the following commands:

- Server commands

  This group of commands includes commands for creating a database cluster and controlling the database. You can run these commands on the server where the database is operating.

  To use these commands, you must configure the environment variables.

See

- Refer to "PostgreSQL Server Applications" under "Reference" in the PostgreSQL Documentation, or "Reference" for information on server commands.
- Refer to "Configure the environment variables" in the procedure to create instances in "Using the initdb Command" in the Installation and Setup Guide for Server for information on configuring the environment variables.

- Client commands

  This group of commands includes the psql command and commands for extracting the database cluster to a script file. These commands can be executed on the client that can connect to the database, or on the server on which the database is running.

  To use these commands, you must configure the environment variables.

See

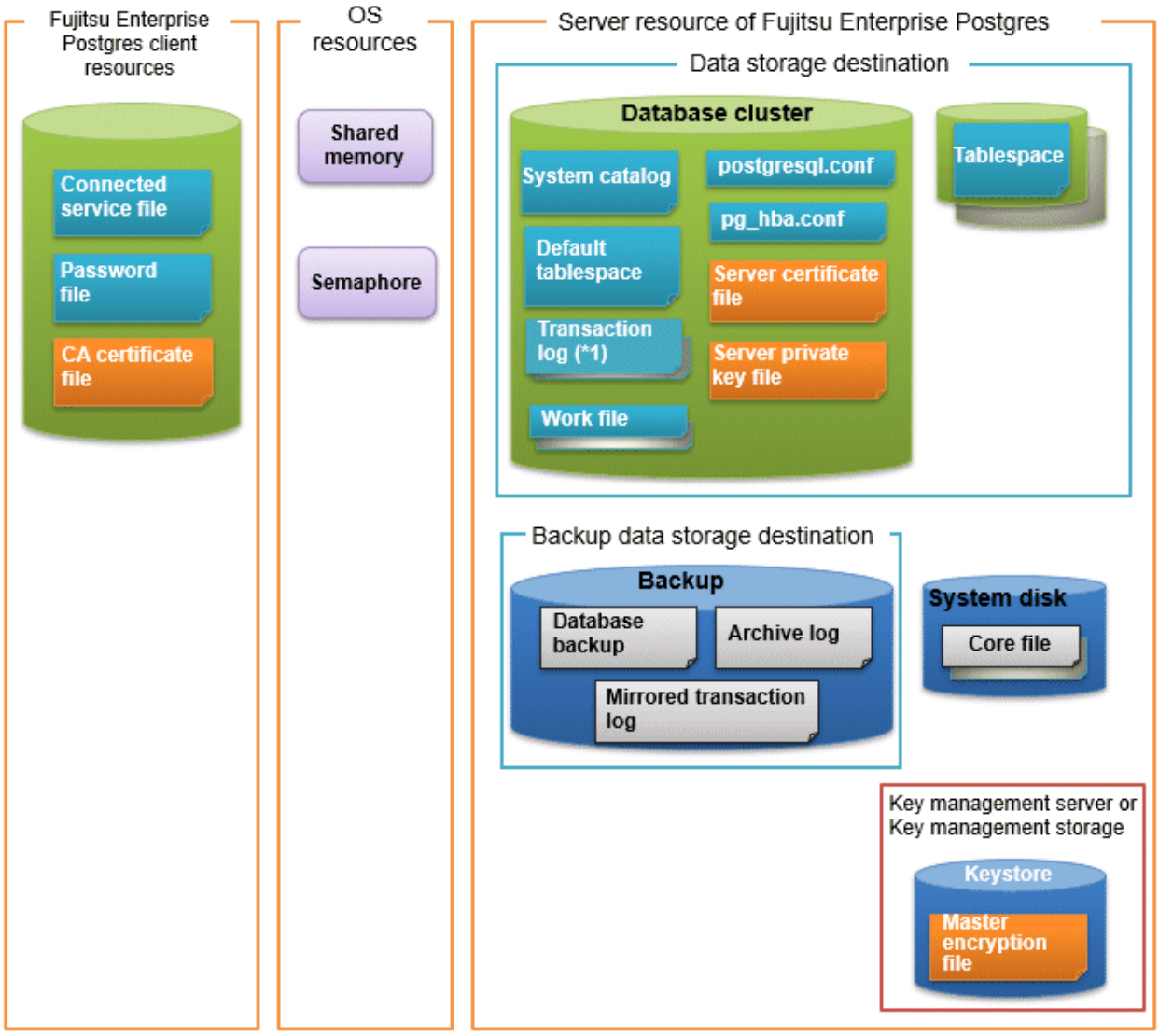
- Refer to "PostgreSQL Client Applications" under "Reference" in the PostgreSQL Documentation, or "Reference" for information on client commands.
- Refer to "Configuring Environment Variables" in the Installation and Setup Guide for Client for information on the values to be set in the environment variables.

1.5 Operating Environment of Fujitsu Enterprise Postgres

This section describes the operating environment and the file composition of Fujitsu Enterprise Postgres.

1.5.1 Operating Environment

The following figure shows the configuration of the Fujitsu Enterprise Postgres operating environment. The tables given below list the roles of the OS resources and Fujitsu Enterprise Postgres resources.



*1: To distribute the I/O load, place the transaction log on a different disk from the data storage destination.

Table 1.1 OS resources

Type	Role
Shared memory	Used when a database process exchanges information with an external process.
Semaphore	

Table 1.2 Fujitsu Enterprise Postgres client resources

Type	Role
Connection service file	Specifies information, such as the host name, user name, and password, for connecting to Fujitsu Enterprise Postgres.
Password file	Securely manages the password for connecting to Fujitsu Enterprise Postgres.
CA certificate file	CA (certificate authority) certificate used for server authentication when encrypting communication data.

Table 1.3 Server resources of Fujitsu Enterprise Postgres

Type	Role
Database cluster	Database storage area on the database storage disk. It is a collection of databases managed by an instance.
System catalog	Contains information required for the system to run, including the database definition information and the operation information created by the user.
Default tablespace	Contains table files and index files stored by default.
Transaction log	Contains log information in case of a crash recovery or rollback. This is the same as the WAL (Write Ahead Log).
Work file	Work file used when executing applications or commands.
postgresql.conf	Contains information that defines the operating environment of Fujitsu Enterprise Postgres.
pg_hba.conf	Fujitsu Enterprise Postgres uses this file to authenticate individual client hosts.
Server certificate file	Contains information about the server certificate to be used when encrypting communication data and authenticating a server.
Server private key file	Contains information about the server private key to be used when encrypting communication data and authenticating a server
Tablespace	Stores table files and index files in a separate area from the database cluster. Specify a space other than that under the database cluster.
Backup	Stores the data required for recovering the database when an error, such as disk failure, occurs.
Database backup	Contains the backup data for the database.
Archive log	Contains the log information for recovery.
Mirrored transaction log (mirrored WAL)	Enables a database cluster to be restored to the state immediately before an error even if both the database cluster and transaction log fail when performing backup/recovery operations using the pgx_dmpall command or WebAdmin.
Core file	Fujitsu Enterprise Postgres process core file that is output when an error occurs during a Fujitsu Enterprise Postgres process.
Key management server or key management storage	Server or storage where the master encryption key file is located.
Master encryption key file	Contains the master encryption key to be used when encrypting storage data. The master encryption key file is managed on the key management server or key management storage.

1.5.2 File Composition

Fujitsu Enterprise Postgres consists of the following files for controlling and storing the database. The table below shows the relationship between the number of such files and their location within a single instance.

Table 1.4 Number of files within a single instance and how to specify their location

File type	Required	Quantity	How to specify the location
Program files	Y	Multiple	Note that "<*x*>" indicates the product version. By default, the following location is used: %Program Files%\Fujitsu\fsepv<*x*>server64
Database cluster	Y	1	Specify using WebAdmin or server commands.

File type	Required	Quantity	How to specify the location
Tablespace	Y	Multiple	Specify a space other than that under the database cluster, using pgAdmin or the DDL statement.
Backup	Y	Multiple	Specify using WebAdmin or server commands.
Core file	Y	Multiple	Specify using WebAdmin, server commands, or postgresql.conf.
Server certificate file (*1)	N	1	Specify using postgresql.conf.
Server private key file (*1)	N	1	Specify using postgresql.conf.
Master encryption key file (*1)	N	1	Specify the directory created as the key store using postgresql.conf.
Connection service file (*1)	N	1	Specify using environment variables.
Password file (*1)	N	1	Specify using environment variables.
CA certificate file (*1)	N	1	Specify using environment variables.

Y: Mandatory
N: Optional

*1: Set manually when using the applicable feature.

Note

If anti-virus software is used, set scan exception settings for directories so that none of the files that comprise Fujitsu Enterprise Postgres are scanned for viruses. Alternatively, if the files that comprise Fujitsu Enterprise Postgres are to be scanned for viruses, stop Fujitsu Enterprise Postgres and perform the scan when tasks that use Fujitsu Enterprise Postgres are not operating.

1.6 Notes on Compatibility of Applications Used for Operations

When you upgrade Fujitsu Enterprise Postgres to a newer version, there may be some effect on applications due to improvements or enhancements in functionality.

Take this into account when creating applications so that you can maintain compatibility after upgrading to a newer version of Fujitsu Enterprise Postgres.

See

Refer to "Notes on Application Compatibility" in the Application Development Guide for details.

1.7 Notes on Upgrading Database Instances

When upgrading FUJITSU Enterprise Postgres 9.4 or newer database instances to FUJITSU Enterprise Postgres 10 or later using pg_upgrade, there are certain steps you need to follow.

Before using pg_upgrade, remove the following extensions from all databases in the instance, except "template0":

- pg_stat_statements
- pgx_io
- pgx_paging

- pgx_network
- pgx_network_err
- pgx_cpu
- pgx_memory
- pgx_swap
- pgx_disk
- pgx_process
- pgx_log
- oracle_compatible
- pg_dbms_stats
- pg_hint_plan

For all databases except "template0", execute the following command to remove these extensions:

```
DROP EXTENSION extensionName;
```

Once the pg_upgrade operation is complete, for all databases except "template0", execute the following command to re-create these extensions as required:

```
CREATE EXTENSION extensionName;
```

- It is strongly recommended to back up the database using pg_dump before performing pg_upgrade or using DROP EXTENSION.
- If there are any columns created in the user tables using a data type from these extensions, then DROP EXTENSION will also drop these columns. Therefore, it is essential that alternate upgrade mechanisms are considered instead of pg_upgrade, in such scenarios. These may include pg_dump/pg_restore.

1.7.1 Additional Steps for upgrading to Fujitsu Enterprise Postgres with Vertical Clustered Index (VCI) Enabled

When upgrading FUJITSU Enterprise Postgres 11.0 or earlier instances that are using the VCI extension to FUJITSU Enterprise Postgres 12 or later using pg_upgrade, additional steps must be performed because of the incompatibility of the VCI extension between FUJITSU Enterprise Postgres 12 or later and FUJITSU Enterprise Postgres 11 or earlier.

Follow the procedure below in all databases in the FUJITSU Enterprise Postgres 11 or earlier instance, except "template0".

Before upgrading

1. Obtain the CREATE INDEX Definitions

   Run the query below to list all the VCI indexes created in the database. Ensure that these indexes are re-created in the FUJITSU Enterprise Postgres 12 or later instance after pg_upgrade has finished.

   ```
   SELECT nspname || '.' || relname AS index_relname,* FROM pg_class, pg_namespace
     WHERE relnamespace = pg_namespace.oid AND relam IN (SELECT oid FROM pg_am WHERE amname='vci');
   ```

   For each index_relname listed above, execute the commands below to obtain the CREATE INDEX definition (to use the same SQL syntax while re-creating the indexes on the FUJITSU Enterprise Postgres 12 or later instance).

   ```
   SELECT pg_get_indexdef('indexName'::regclass);
   ```

2. Drop the VCI indexes and VCI extension along with all its dependencies.

   To remove all the VCI indexes and VCI internal objects that are created in Fujitsu Enterprise Postgres, execute the commands below. VCI internal objects will be created in FUJITSU Enterprise Postgres 12 or later automatically when CREATE EXTENSION for VCI is executed.

   ```
   DROP EXTENSION VCI CASCADE;
   ```

   Note

   To restore the VCI extension in the FUJITSU Enterprise Postgres 11 or earlier instance, execute CREATE EXTENSION.

After upgrading

Once the pg_upgrade operation is complete, for all databases except "template0", execute CREATE EXTENSION to create the VCI extension, and then execute CREATE INDEX for all the VCI indexes as required.

Chapter 2 Starting an Instance and Creating a Database

This chapter describes basic operations, from starting an instance to creating a database.

2.1 Starting and Stopping an Instance

This section describes how to start and stop an instance.

- 2.1.1 Using WebAdmin
- 2.1.2 Using Commands

Point

To automatically start or stop an instance when the operating system on the database server is started or stopped, refer to "Configuring Automatic Start and Stop of an Instance" in the Installation and Setup Guide for Server and configure the settings.

Note

The collected statistics are initialized if an instance is stopped in the "Immediate" mode or if it is abnormally terminated. To prepare for such initialization of statistics, consider regular collection of the statistics by using the SELECT statement. Refer to "The Statistics Collector" in "Server Administration" in the PostgreSQL Documentation for information on the statistics.

2.1.1 Using WebAdmin

WebAdmin enables you to start or stop an instance and check its operating status.

Starting an instance

Start an instance by using the [Instances] tab in WebAdmin.

is displayed when an instance is stopped.

To start a stopped instance, click .

Stopping an instance

Stop an instance by using the [Instances] tab in WebAdmin.

 is displayed when an instance is active.

To stop an active instance, click .

Stop mode

Select the mode in which to stop the instance. The following describes the operations of the modes:

Stop mode	Connected clients	Backup being executed using the command
Smart mode (*1)	Waits for all connected clients to be disconnected.	Waits for backups being executed using the command to finish.
Fast mode	Rolls back all transactions being executed and forcibly disconnects clients.	Terminates backups being executed using the command.
Immediate mode	All server processes are terminated immediately. Crash recovery is executed the next time the instance is started.	

Stop mode	Connected clients	Backup being executed using the command
Kill process mode	Send SIGKILL to the process and abort all active transactions. This will lead to a crash-recovery run at the next restart.	

*1: When the processing to stop the instance in the Smart mode has started and you want to stop immediately, use the following procedure:

1. Restart the Web server feature of WebAdmin.
2. In the [Instances] tab, click .
3. In the [Instances] tab, click , and select the Immediate mode to stop the instance.

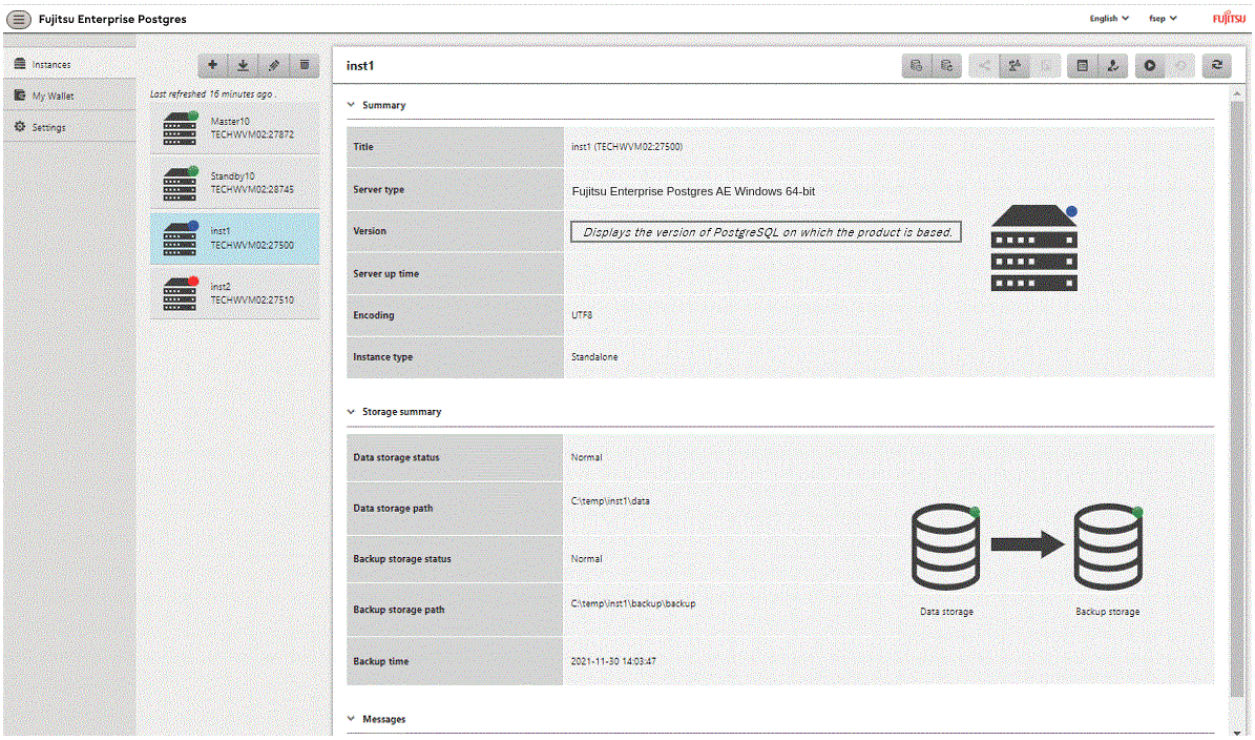
Checking the operating status of an instance

You can check the operating status of an instance by using the [Instances] tab. The following indicators are used to show the status of a resource.

Status indicator	Explanation
	The resource is operating normally.
	The resource is stopped.
	There is an error in the resource.
	An operation is in progress on this resource or the status is being checked.
	The resource is not operating optimally and needs intervention.

If an instance stops abnormally, remove the cause of the stoppage and start the instance by using WebAdmin.

Figure 2.1 Example of operating status indicators



Note

- When operating WebAdmin, click to update the status. WebAdmin will reflect the latest status of the operation or the instance resources from the server.
- If an error occurs while communicating with the server, there may be no response from WebAdmin. When this happens, close the browser and then log in again. If this does not resolve the issue, check the event log of the server and confirm whether a communication error has occurred.
- The following message is output during startup of an instance when the startup process is operating normally, therefore, the user does not need to be aware of this message:

```
FATAL:  the database system is starting up
```

2.1.2 Using Commands

The Windows service-related commands enable you to start or stop an instance and to check its operating state.

If you are to use Windows services, you should register instances in Windows services.

See

Refer to "When an instance was created with WebAdmin" in "Configuring Automatic Start and Stop of an Instance" in the Installation Guide for Server for information on registering instances in Windows services.

Note

While it is also possible for you to execute the pg_ctl command to start and stop instances without having to register instances in Windows services, it is recommended that you use Windows services to start and stop instances for the following reason:

- If you use the pg_ctl command to start an instance, the instance will be started as a user process. Therefore, when you close the [Command Prompt] window in which you executed the command, Windows forces the postgres process to stop.

Starting an instance

You can start an instance by specifying the service name in the net start command or sc start command.

Also, you can use the following procedure to start an instance in the Windows services window:

1. Display the [Services] window

   In the [Administrative Tools], click [Services].

2. Start a service

   Select the instance name that you wish to start from the services list, and click [Start Service].

Stopping an instance

You can stop an instance by specifying the service name in the net stop command or sc stop command.

Also, you can use the following procedure to stop an instance in the Windows services window:

1. Display the [Services] window

   In the [Administrative Tools], click [Services].

2. Stop the service

   Select the instance name that you wish to stop from the services list, and click [Stop Service]. If you stop a service while applications and commands are running, Fujitsu Enterprise Postgres will force those applications and commands to close and will stop normally.

Checking the operating state of an instance

Use the following procedure to check if an instance is operating correctly immediately after performing the operation to start an instance:

1. Display the [Services] window

   In the [Start] menu, select [Administrative Tools], and then click [Services].

2. Check the state of the service

   In the services list, check the state of the services for the applicable Fujitsu Enterprise Postgres.

To check the operating state of an instance during operation, use the pg_ctl command.

Specify the following in the pg_ctl command:

- Specify "status" as the mode.
- Specify the data storage destination directory in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

Example

When the instance is active:

```
> pg_ctl status -D D:\database\inst1
pg_ctl: server is running (PID: 1234)
```

When the instance is inactive:

```
> pg_ctl status -D D:\database\inst1
pg_ctl: no server running
```

Information

You can also use the net start command or sc query command to check the operating state of an instance.

See

Refer to "pg_ctl" in "Reference" in the PostgreSQL Documentation for information on the pg_ctl command.

2.2 Creating a Database

This section explains how to create a database.

- 2.2.1 Using pgAdmin
- 2.2.2 Using Client Commands

2.2.1 Using pgAdmin

Follow the procedure below to define a database using pgAdmin.

1. In the pgAdmin window, right-click [Databases] in the [Browser] pane, and then click [Create] >> [Database...] to display a [Create - Database] window.
2. Specify appropriate values for the following items in the [Create - Database] window.
   - [General] tab
     - [Database]: Name of the database to be managed

3. Click [Save] to create the database.

2.2.2 Using Client Commands

Follow the procedure below to define a database using client commands.

An example of operations on the server is shown below.

1. Use psql command to connect to the postgres database.
   Execute psql postgres.

```
> psql postgres
psql (<x>) (*1)
Type "help" for help.
```

   *1: *<x>* indicates the PostgreSQL version on which this product is based.

2. Create the database.
   To create the database, execute the CREATE DATABASE databaseName; statement.

```
postgres=# CREATE DATABASE db01;
CREATE DATABASE
```

3. Confirm that the database is created.
   Execute \l+, and confirm that the name of the database created in step 2 is displayed.

```
postgres=# \l+
```

4. Disconnect from the postgres database.
   Execute \q to terminate the psql command.

```
postgres=# \q
```

You can create a database using the createdb command.

See

Refer to "Creating a Database" in "Tutorial" in the PostgreSQL Documentation for information on creating a database using the createdb command.

Chapter 3 Backing Up the Database

This chapter describes how to back up the database.

Backup methods

The following backup methods enable you to recover data to a backup point or to the state immediately preceding disk physical breakdown or data logical failure.

- Backup using WebAdmin

  This method enables you to back up data through intuitive window operations using the GUI.

  WebAdmin is used for recovery.

- Backup using the pgx_dmpall command

  Execute the pgx_dmpall command with a script to perform automatic backup.

  To back up data automatically, you must register the process in the automation software of the operating system. Follow the procedure given in the documentation for your operating system.

  The pgx_rcvall command is used for recovery.

Information

By using a copy command created by the user, the pgx_dmpall command and the pgx_rcvall command can back up database clusters and tablespaces to any destination and recover them from any destination using any copy method. Refer to "Chapter 14 Backup/Recovery Using the Copy Command" for details.

Approximate backup time

The formula for deriving the approximate backup time when you use WebAdmin or the pgx_dmpall command is as follows:

```
backupTime = dataStorageDestinationUsage / diskWritePerformance x 1.5
```

- *dataStorageDestinationUsage*: Disk usage at the data storage destination
- *diskWritePerformance*: Maximum data volume (bytes/second) that can be written per second in the system environment where operation is performed
- 1.5: Coefficient to factor in tasks other than disk write (which is the most time-consuming step)

If using the copy command with the pgx_dmpall command, the backup time will depend on the implementation of the copy command.

Note

- Backup operation cannot be performed on an instance that is part of a streaming replication cluster in standby mode.
- Use the selected backup method continuously.

  There are several differences, such as the data format, across the backup methods. For this reason, the following restrictions apply:

  - It is not possible to use one method for backup and another for recovery.
  - It is not possible to convert one type of backup data to a different type of backup data.
- Mirrored WALs can be used only for backup/recovery using the pgx_dmpall command or WebAdmin.
- There are several considerations for the backup of the keystore and backup of the database in case the data stored in the database is encrypted. Refer to the following for details:

  - 5.6.4 Backing Up and Recovering the Keystore
  - 5.7 Backing Up and Restoring/Recovering the Database

- If you have defined a tablespace, back it up. If you do not back it up, directories for the tablespace are not created during recovery, which may cause the recovery to fail. If the recovery fails, refer to the event log, create the tablespace, and then perform the recovery process again.
- If performing backups with WebAdmin, the following password file is temporarily created during backup for WebAdmin to connect to the database:
  - *userProfileFolder**localSettingsFolder*\Fujitsu\fsep_*version**instanceName*\pgpass.conf

  Therefore, when you are backing up corefiles created in the core_directory parameter of postgresql.conf, or log files created in the log_directory parameter of postgresql.conf, ensure not to back up the password files located in the same directories at the same time.

Information

The following methods can also be used to perform backup. Performing a backup using these methods allows you to restore to the point when the backup was performed.

- Backup using an SQL-based dump

  Dump the data by using SQL. This backup method also enables data migration.
- File system level backup

  This backup method requires you to stop the instance and use OS commands to backup database resources as files.
- Backup by continuous archiving

  This is the standard backup method for PostgreSQL.

Refer to "Backup and Restore" in "Server Administration" in the PostgreSQL Documentation for information on these backup methods.

3.1 Periodic Backup

It is recommended that you perform backup periodically.

Backing up data periodically using WebAdmin or the pgx_dmpall command has the following advantages:

- This method reduces disk usage, because obsolete archive logs (transaction logs copied to the backup data storage destination) are deleted. It also minimizes the recovery time when an error occurs.

Backup cycle

The time interval when backup is performed periodically is called the backup cycle. For example, if backup is performed every morning, the backup cycle is 1 day.
The backup cycle depends on the jobs being run, but on Fujitsu Enterprise Postgres it is recommended that operations are run with a backup cycle of at least once per day.

3.2 Backup Methods

This section describes the methods for backing up the database.

- 3.2.1 Using WebAdmin
- 3.2.2 Using Server Commands

3.2.1 Using WebAdmin

You can use WebAdmin to perform backup and check the backup status.

Note

- If backup is disabled for an instance, you will not be able to back up or restore the instance. Refer to "[Backup]" in "Creating an Instance" in the Installation and Setup Guide for Server for details.
- If the data to be stored in the database is to be encrypted, it is necessary to enable the automatic opening of the keystore before doing so. Refer to "5.6.3 Enabling Automatic Opening of the Keystore" for details.
- WebAdmin uses the labels "Data storage path", "Backup storage path" and "Transaction log path" to indicate "data storage destination", "backup data storage destination" and "transaction log storage destination" respectively. In this manual these terms are used interchangeably.

Backup operation

Follow the procedure below to back up the database.

1. Select the database to back up

   In the [Instances] tab, select the instance to be backed up and click .

2. Back up the database

   The [Backup] dialog box is displayed. To perform backup, click [Yes].
   An instance is automatically started when backup is performed.

Backup status

If an error occurs and backup fails, [Error] is displayed adjacent to [Data storage status] or [Backup storage status] in the [Instances] tab. An error message is also displayed in the message list.

In this case, the backup data is not optimized. Ensure that you check the backup result whenever you perform backup. If backup fails, [Solution] appears to the right of the error message. Clicking this button displays information explaining how to resolve the cause of the error. Remove the cause of failure, and perform backup again.

3.2.2 Using Server Commands

Use the pgx_dmpall command and pgx_rcvall command to perform backup and check the backup result.

Preparing for backup

You must prepare for backup before actually starting the backup process.

Follow the procedure below.

See

Refer to "Preparing Directories to Deploy Resources" in the Installation and Setup Guide for Server for information on the location of directories required for backup and for points to take into account.

1. Prepare the backup data storage disk

   For backup, prepare a separate disk unit from the database storage disk and mount it using the operating system commands.

2. Create a directory where the backup data will be stored

   Create an empty directory.

   In [Properties] in Windows(R) Explorer, set appropriate permissions so that only the instance administrator can access the directory.

See

Refer to [Help and Support] in Windows(R) for information on [Properties].

3. Specify the settings required for backup

   Stop the instance, and set the following parameters in the postgresql.conf file.

   Start the instance after editing the postgresql.conf file.

Parameter name	Setting	Description
backup_destination	Name of the directory where the backup data will be stored	Specify the name of the directory where the backup data will be stored. Appropriate privileges that allow only the instance administrator to access the directory must already be set. Place the backup data storage destination directory outside the data storage destination directory, the tablespace directory, and the transaction log storage destination directory.
archive_mode	on	Specify the archive log mode. Specify [on] (execute).
archive_command	'cmd /c ""*installationDirectory*\\bin\ \pgx_walcopy.cmd" "%p" "*backupDataStorageDestinationDirectory*\ \archived_wal\\%f"""	Specify the path name of the command that will save the transaction log and the storage destination. Note the following when specifying the path: - Specify \\ as the path delimiter. - Enclose the path in double quotes ("") if it contains spaces.
archive_library	"(default)	Specify the archive library. Specify ["](default).

   Refer to "Appendix A Parameters" and "Write Ahead Log" under "Server Administration" in the PostgreSQL Documentation for information on the parameters.

Backup operation (file backup)

Use the pgx_dmpall command to perform file backup. You can even embed the pgx_dmpall command in OS automation software to perform backup.

The backup data is stored in the directory specified in the backup_destination parameter of postgresql.conf.

Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

Example

```
> pgx_dmpall -D D:\database\inst1
```

Note

Backup stores the data obtained during the backup and the backup data of the data obtained during previous backup.

If the data to be stored in the database is encrypted, refer to the following and back up the keystore:

- 5.6.4 Backing Up and Recovering the Keystore

Backup status

Use the pgx_rcvall command to check the backup status.

Specify the following values in the pgx_rcvall command:

- The -l option indicates backup data information.
- Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

```
> pgx_rcvall -l -D D:\database\inst1
Date                    Status          Dir
2022-03-01 13:30:40     COMPLETE        E:/backup/inst1/2022-03-01_13-30-40
```

If an error occurs and backup fails, a message is output to the event log.

In this case, the backup data is not optimized. Ensure that you check the backup result whenever you perform backup. If backup fails, remove the cause of failure and perform backup again.

See

Refer to "pgx_dmpall" and "pgx_rcvall" in the Reference for information on the pgx_dmpall command and pgx_rcvall command.

Setting a restore point

In case you want to recover your database to a certain point in time, you can name this particular point in time, which is referred to as the restore point, by using the psql command.

By setting a restore point before executing an application, it becomes easy to identify up to which point in time the data will be reverted.

A restore point can be set to any point in time after a backup is executed. However, if a restore point is set before a backup is executed, the database cannot be recovered to that point in time. This is because restore points are recorded in the archive logs, and the archive logs are discarded when backups are executed.

Example

The following example uses the psql command to connect to the database and execute the SQL statement to set a restore point.

However, when considering continued compatibility of applications, do not use functions directly in SQL statements. Refer to "Notes on Application Compatibility" in the Application Development Guide for details.

```
postgres=# SELECT pg_create_restore_point('batch_20220303_1');
LOG:  restore point "batch_20220303_1" created at 0/20000E8
STATEMENT:  select pg_create_restore_point('batch_20220303_1');
 pg_create_restore_point
-------------------------
 0/20000E8
(1 row)
```

Refer to "15.3.2 Using the pgx_rcvall Command" for information on using a restore point to recover the database.

Note

- Name restore points so that they are unique within the database. Add the date and time of setting a restore point to distinguish it from other restore points, as shown below:
  - YYMMDD_HHMMSS
    - YYMMDD: Indicates the date
    - HHMMSS: Indicates the time
- There is no way to check restore points you have set. Keep a record in, for example, a file.

See

Refer to "System Administration Functions" under "Functions and Operators" in the PostgreSQL Documentation for information on pg_create_restore_point.

Chapter 4 Configuring Secure Communication Using Secure Sockets Layer

If communication data transferred between a client and a server contains confidential information, encrypting the communication data can protect it against threats, such as eavesdropping on the network.

4.1 Configuring Communication Data Encryption

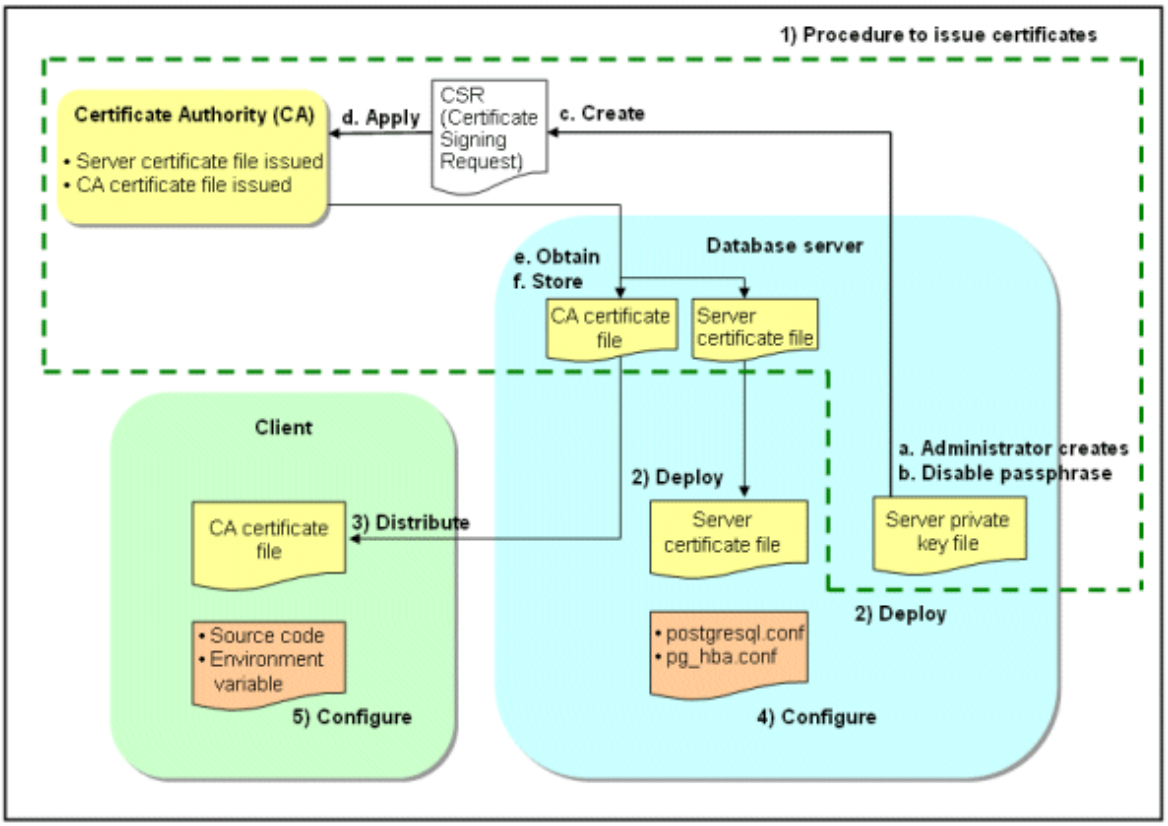
To encrypt communication data transferred between a client and a server, configure communication data encryption as described below. Communication data encryption not only protects the communication content, but it also guards against man-in-the-middle (MITM) attacks (for example, data and password theft through server impersonation).

Table 4.1 Configuration procedure

Configuration procedure
1) Issue a certificate
2) Deploy a server certificate file and a server private key file
3) Distribute a CA certificate file to the client
4) Configure the operating environment for the database server
5) Configure the operating environment for the client

The following figure illustrates the environment for communication data encryption.

Figure 4.1 Environment for communication data encryption



4.1.1 Issuing a Certificate

For authenticating servers, you must acquire a certificate issued by the certificate authority (CA).
Fujitsu Enterprise Postgres supports X.509 standard PEM format files. If the certificate authority issues a file in DER format, use a tool such as the openssl command to convert the DER format file to PEM format.

The following provides an overview of the procedure. Refer to the procedure published by the public or independent certificate authority (CA) that provides the certificate file for details.

a. Create a server private key file

b. Disable the passphrase for the server private key file

c. Create a CSR (signing request for obtaining a server certificate) from the server private key file

d. Apply to the certificate authority (CA) for a server certificate

e. Obtain a server certificate file and a CA certificate file from the certificate authority (CA)

f. Store the server certificate file and the CA certificate file
   Note: If you lose or destroy the certificates, you will need to have them re-issued.

The above procedure enables you to prepare the following files:

- Server private key file
- Server certificate file
- CA certificate file

4.1.2 Deploying a Server Certificate File and a Server Private Key File

Create a directory on the local disk of the database server and store the server certificate file and the server private key file in it.
Use the operating system features to set access privileges for the server certificate file and the server private key file so that only the database administrator has load privileges.
Back up the server certificate file and the server private key file in the event that data corruption occurs and store them securely.

4.1.3 Distributing a CA Certificate File to the Client

Create a directory on the local disk of the client and place the distributed CA certificate file there. Use the operating system features to set load privileges to protect the CA certificate file against accidental deletion.

4.1.4 Configuring the Operating Environment for the Database Server

See

Refer to "Secure TCP/IP Connections with SSL" under "Server Administration" in the PostgreSQL Documentation for details.

4.1.5 Configuring the Operating Environment for the Client

See

Refer to the following sections in the Application Development Guide for details, depending on your application development environment:

- "Settings for Encrypting Communication Data" under "Setup" in "JDBC Driver"
- "Settings for Encrypting Communication Data" under "Setup" in "C Library (libpq)"
- "Settings for Encrypting Communication Data" under "Setup" in "Embedded SQL in C"

4.1.6 Performing Database Multiplexing

When you perform communication that uses database multiplexing and a Secure Socket Layer server certificate, certificates with the same "Common Name" must be used. To ensure this, take one of the following actions:

- Create one server certificate, replicate it, and place a copy on each server used for database multiplexing.
- Create a server certificate with the same "Common Name" for each server used for database multiplexing.

See

Refer to "Using the Application Connection Switch Feature" in the Application Development Guide for information on how to specify applications on the client.

Chapter 5 Protecting Storage Data Using Transparent Data Encryption

This chapter describes how to encrypt data to be stored in the database.

5.1 Protecting Data Using Encryption

With PostgreSQL, data in a database is protected from access by unauthorized database users through the use of authentication and access controls. However, the OS file is not protected from attackers who bypass the database server's authentication and access controls.

With Fujitsu Enterprise Postgres, data inside the OS file is encrypted, so valuable information is protected even if the file or disk is stolen.

Data to be stored in a database is encrypted when it is written to the data file, and decrypted when it is read.

This is performed automatically by the instance, so the user and the application need not be aware of key management and encryption or decryption. This process is called TDE (Transparent Data Encryption).

The characteristics of TDE are described below.

Encryption mechanisms

Two-layer encryption key and the keystore

In each tablespace, there is a tablespace encryption key that encrypts and decrypts all the data within. The tablespace encryption key is encrypted by the master encryption key and saved.
Only one master encryption key exists in a database cluster. It is encrypted based on a passphrase specified by the user and stored in a keystore. Fujitsu Enterprise Postgres provides a file-based keystore. Attackers who do not know the passphrase cannot read the master encryption key from the keystore.

Strong encryption algorithms

TDE uses the Advanced Encryption Standard (AES) as its encryption algorithm. AES was adopted as a standard in 2002 by the United States Federal Government, and is used throughout the world.

Faster encryption and decryption based on hardware

TDE minimizes the overhead of encryption and decryption by using the AES-NI (Advanced Encryption Standard New Instructions) built into Intel(R) Xeon(R) processors since the 5600 series. This means that even in situations where previously the minimum encryption target was selected as a tradeoff between performance and security, it is now possible to encrypt all the data of an application.

Refer to the Intel Corporation's website for information on the list of processors equipped with AES-NI.

Zero overhead storage areas

Encryption does not change the size of data stored in tables, indexes, or WAL. There is, therefore, no need for additional estimates or disks.

Scope of encryption

All user data within the specified tablespace

The tablespace is the unit for specifying encryption. All tables, indexes, temporary tables, and temporary indexes created in the encrypted tablespace are encrypted. There is no need for the user to consider which tables and strings to encrypt.

Refer to "5.4 Encrypting a Tablespace" for details.

Backup data

The pgx_dmpall command and pg_basebackup command create backup data by copying the OS file. Backups of the encrypted data are, therefore, also encrypted. Information is protected from leakage even if the backup medium is stolen.

WAL and temporary files

WAL, which is created by updating encrypted tables and indexes, is encrypted with the same security strength as the update target. When large merges and sorts are performed, the encrypted data is written to a temporary file in encrypted format.

Streaming replication support

You can combine streaming replication and transparent data encryption. The data and WAL encrypted on the primary server is transferred to the standby server in its encrypted format and stored.

Note

The following are not encrypted:

- pg_dump and pg_dumpall output files
- Files output by the COPY command
- Notification event payloads that communicate using the LISTEN or NOTIFY command
- Checksum validation is not performed on encrypted tablespaces during backup and when using the pg_checksum utility.

5.2 Setting the Master Encryption Key

To use transparent data encryption, you must create a keystore and set the master encryption key.

1. In the keystore_location parameter of postgresql.conf, specify the directory to store the keystore.

   Specify a different location for each database cluster.

   ```
   keystore_location = 'C:\\key\\store\\location'
   ```

   Refer to "Appendix A Parameters" for information on postgresql.conf.

   After editing the postgresql.conf file, either start or restart the instance.

   - Using WebAdmin

     Refer to "2.1.1 Using WebAdmin", and restart the instance.

   - Using commands

     Refer to "2.1.2 Using Commands", and restart the instance.

2. Execute an SQL function, such as the one below, to set the master encryption key. This must be performed by the superuser. Execute it as the database superuser.

   ```
   SELECT pgx_set_master_key('passphrase');
   ```

   The value "passphrase" is the passphrase that will be used to open the keystore. The master encryption key is protected by this passphrase, so avoid specifying a short simple string that is easy to guess.

   Refer to "B.2 Transparent Data Encryption Control Functions" for information on the pgx_set_master_key function.

Note

Note that if you forget the passphrase, you will not be able to access the encrypted data. There is no method to retrieve a forgotten passphrase and decrypt data. Do not, under any circumstances, forget the passphrase.

The pgx_set_master_key function creates a file with the name keystore.ks in the keystore storage destination. It also creates a master encryption key from random bit strings, encrypts it with the specified passphrase, and stores it in keystore.ks. At this point, the keystore is open.

5.3 Opening the Keystore

To create encrypted tablespaces and access the encrypted data, you must first open the keystore. When you open the keystore, the master encryption key is loaded into the database server memory and becomes usable for encryption and decryption.

You need to open the keystore each time you start the instance. To open the keystore, the database superuser must execute the following SQL function.

```
SELECT pgx_open_keystore('passphrase');
```

The value "passphrase" is the passphrase specified during creation of the keystore.

Refer to "B.2 Transparent Data Encryption Control Functions" for information on the pgx_open_keystore function.

Note that, in the following cases, the passphrase must be entered when starting the instance, because the encrypted WAL must be decrypted for recovery. In this case, the above-mentioned pgx_open_keystore function cannot be executed.

- If performing crash recovery at the time of starting the instance
- If performing recovery using continuous archiving

For the above cases, select one of the following methods:

- Use an automatically opening keystore

  Select this method if ease of operation has priority over enhanced security. When using an automatically opening keystore, the content of the keystore file is decrypted and a copy of the keystore file is generated. Although the content of this file is obfuscated, the level of security becomes slightly weaker.

  Select this method if performing operations using WebAdmin.

- Enter a passphrase when starting an instance

  Select this method if enhanced security has priority over ease of operation.

  Specify the --keystore-passphrase in the pg_ctl command and start the instance. This displays the prompt that asks for the passphrase to be entered.

  ```
  > pg_ctl --keystore-passphrase start
  Enter the passphrase:
  The server is currently initiating
  >
  ```

  After performing the above operation, use the pg_ctl command to stop the instance.

  Then start the instance in Windows services. Refer to "2.1.2 Using Commands" for information on how to start an instance in Windows services.

Point

When using an automatically opening keystore, you do not need to enter the passphrase and you can automatically open the keystore when the database server starts. Refer to "5.6.3 Enabling Automatic Opening of the Keystore" for details.

5.4 Encrypting a Tablespace

The keystore must be open before you can create an encrypted tablespace.

When creating a tablespace that will be encrypted, configure the encryption algorithm in the runtime parameters. For example, to create a tablespace with the name secure_tablespace using AES with a key length of 256 bits as the encryption algorithm, configure as shown below.

```
-- Specify the encryption algorithm for the tablespace to be created below
SET tablespace_encryption_algorithm = 'AES256';
CREATE TABLESPACE secure_tablespace LOCATION 'C:\My\Data\Dir';
-- Specify that the tablespace to be created below is not to be encrypted
SET tablespace_encryption_algorithm = 'none';
```

Or

```
CREATE TABLESPACE secure_tablespace LOCATION '\My\Data\Dir' WITH (tablespace_encryption_algorithm =
'AES256' );
```

When the tablespace is empty, the encryption algorithm can be modified with the command below.

```
ALTER TABLESPACE secure_tablespace SET (tablespace_encryption_algorithm=AES256);
```

Trying to set the encryption algorithm for a non-empty tablespace causes an error.

You can use AES with a key length of 128 bits or 256 bits as the encryption algorithm. It is recommended that you use 256-bit AES. Refer to "Appendix A Parameters" for information on how to specify the runtime parameters.

If user provides both GUC and command line options while creating the tablespace, the preference is given to the command line option.

The pg_default and pg_global tablespaces cannot be encrypted.

Create tables and indexes in the encrypted tablespace that you created. Relations created in the encrypted tablespace are automatically encrypted.

Example

Example 1: Specifying an encrypted tablespace when creating it

```
CREATE TABLE my_table (...)
    TABLESPACE secure_tablespace;
```

Example 2: Not explicitly specifying a tablespace when creating it and instead using the default tablespace

```
SET default_tablespace = 'secure_tablespace';
CREATE TABLE my_table (...);
```

The process is the same for encrypting temporary tables and temporary indexes. In other words, either explicitly specify the TABLESPACE clause or list encrypted tablespaces in the temp_tablespaces parameter, and then execute CREATE TEMPORARY TABLE or CREATE INDEX.

Point

If an encrypted tablespace is specified in the TABLESPACE clause of the CREATE DATABASE statement, relations created in the database without explicitly specifying a tablespace will be encrypted. Furthermore, the system catalog will also be encrypted, so the source code of user-defined functions is also protected.

Example: Specifying a tablespace in a database definition statement

```
CREATE DATABASE DB01 TABLESPACE=SP01 ... ;
```

Part of the data is also stored in the system catalog - to encrypt this data as well, specify an encrypted tablespace as above and create a database.

Note

An encrypted tablespace cannot be created from the window used for creating the pgAdmin tablespace. To create an encrypted tablespace, create an encrypted tablespace in the Query tool.

5.5 Checking an Encrypted Tablespace

The pgx_tablespaces system view displays information about whether each tablespace has been encrypted, and about the encryption algorithm. Refer to "C.1 pgx_tablespaces" for information on strings.

You can discover which tablespaces have been encrypted by executing the following SQL statements.

However, when considering continued compatibility of applications, do not reference system catalogs (pg_tablespace) directly in SQL statements.

```
SELECT spcname, spcencalgo
FROM pg_tablespace ts, pgx_tablespaces tsx
WHERE ts.oid = tsx.spctablespace;
```

Example

```
postgres=# SELECT spcname, spcencalgo FROM pg_tablespace ts, pgx_tablespaces tsx WHERE ts.oid =
tsx.spctablespace;
      spcname      | spcencalgo
-------------------+------------
 pg_default        | none
 pg_global         | none
 secure_tablespace | AES256
(3 rows)
```

See

Refer to "Notes on Application Compatibility" in the Application Development Guide for information on how to maintain application compatibility.

5.6 Managing the Keystore

This section describes how to manage the keystore and the master encryption key to guard against the threat of theft.

5.6.1 Changing the Master Encryption Key

Using the same encryption key for an extended period gives attackers an opportunity to decipher the encrypted data. It is recommended that you change the key at regular intervals, or whenever the key is exposed to risk.

Adhere to the industry's best practices for encryption algorithms and key management when considering how often the key should be changed. For example, the NIST in the United States has published "NIST Special Publication 800-57". The PCI DSS also refers to this publication. This publication recommends changing the master encryption key once a year.

To change the master encryption key, execute the pgx_set_master_key function, which is the same function used for configuring the key. Refer to "5.2 Setting the Master Encryption Key" for details.

After changing the master encryption key, you must immediately back up the keystore.

5.6.2 Changing the Keystore Passphrase

In security policies for organizations, it is usually a requirement that the passphrase be changed whenever a security administrator who knows the passphrase is removed from duties due to transfer or retirement. It is also recommended that the passphrase be changed if it is ever exposed to risks due to deception such as social engineering.

To change the keystore passphrase, execute the following SQL function as a superuser.

```
SELECT pgx_set_keystore_passphrase('oldPassphrase', 'newPassphrase');
```

After changing the passphrase, you must immediately back up the keystore.

Refer to "B.2 Transparent Data Encryption Control Functions" for information on the pgx_set_keystore_passphrase function.

5.6.3 Enabling Automatic Opening of the Keystore

When using an automatically opening keystore, you do not need to enter the passphrase and you can automatically open the keystore when the instance starts. Execute the pgx_keystore command to enable automatic opening of the keystore.

```
> pgx_keystore --enable-auto-open C:\key\store\location\keystore.ks
Enter the passphrase:
Automatic opening of the keystore is now enabled
>
```

See

Refer to "pgx_keystore" in the Reference for information on pgx_keystore command.

When automatic opening is enabled, an automatically opening keystore is created in the same directory as the original keystore. The file name of the automatically opening keystore is keystore.aks. The file keystore.aks is an obfuscated copy of the decrypted content of the keystore.ks file. As long as this file exists, there is no need to enter the passphrase to open the keystore when starting the instance.

Do not delete the original keystore file, keystore.ks. It is required for changing the master encryption key and the passphrase. When you change the master encryption key and the passphrase, keystore.aks is recreated from the original keystore file, keystore.ks.

Protect keystore.ks, keystore.aks, and the directory that stores the keystore so that only the user who starts the instance can access them.

Configure the permission of the files so that only the user who starts the instance can access the SQL functions and commands that create these files. Accordingly, manually configure the same permission mode if the files are restored.

Set the permission mode in [Properties] in Windows(R) Explorer.

See

Refer to [Help and Support] in Windows(R) for information on [Properties].

An automatically opening keystore will only open on the computer where it was created.

To disable automatic opening of the keystore, delete keystore.aks.

Note

- To use WebAdmin for recovery, you must enable automatic opening of the keystore.
- Refer to "5.7 Backing Up and Restoring/Recovering the Database" after enabling or reconfiguring encryption to back up the database.
- Specify a different directory from those below as the keystore storage destination:
  - Data storage destination
  - Tablespace storage destination
  - Transaction log storage destination
  - Backup data storage destination

5.6.4 Backing Up and Recovering the Keystore

Back up the keystore at the following times in case it is corrupted or lost. Note that you must store the database and the keystore on separate data storage media. Storing both on the same data storage medium risks the danger of the encrypted data being deciphered if the medium is stolen. A passphrase is not required to open an automatically opening keystore, so store this type of keystore in a safe location.

- When the master encryption key is first configured
- When the master encryption key is changed
- When the database is backed up
- When the keystore passphrase is changed

Point

Do not overwrite an old keystore when backing up a keystore. This is because during database recovery, you must restore the keystore to its state at the time of database backup. When the backup data of the database is no longer required, delete the corresponding keystore.

Example

- Back up the database and the keystore on March 1, 2022.

```
> pgx_dmpall -D D:\database\inst1
> copy C:\key\store\location\keystore.ks C:\keybackup\keystore_20220301.ks
```

  Specify the following in the pgx_dmpall command:

  - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

- Change the master encryption key, and back up the keystore on March 5, 2022.

```
> psql -c "SELECT pgx_set_master_key('passphrase')" postgres
> copy C:\key\store\location\keystore.ks C:\keybackup\keystore_20220305.ks
```

  Specify the following in the psql command:

  - Specify the SQL function that sets the master encryption key in the -c option.
  - Specify the name of the database to be connected to as the argument.

If the keystore is corrupted or lost, restore the keystore containing the latest master encryption key. If there is no keystore containing the latest master encryption key, restore the keystore to its state at the time of database backup, and recover the database from the database backup. This action recovers the keystore to its latest state.

Example

- Restore the keystore containing the latest master encryption key as of March 5, 2022.

```
> copy C:\keybackup\keystore_20220305.ks C:\key\store\location\keystore.ks
```

- If there is no backup of the keystore containing the latest master encryption key, recover the keystore by restoring the keystore that was backed up along with the database on 1 March 2022.

```
> copy C:\keybackup\keystore_20220301.ks C:\key\store\location\keystore.ks
> pgx_rcvall -B E:\backup\inst1 -D D:\database\inst1 --keystore-passphrase
```

  Specify the following in the pgx_rcvall command:

  - Specify the data storage directory in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
  - Specify the backup data storage directory in the -B option.
  - The --keystore-passphrase option prompts you to enter the passphrase to open the keystore.

If you have restored the keystore, repeat the process of enabling automatic opening of the keystore. This ensures that the contents of the automatically opening keystore (keystore.aks) are identical to the contents of the restored keystore.

It is recommended that you do not back up the automatically opening keystore file, keystore.aks. If the database backup medium and the backup medium storing the automatically opening keystore are both stolen, the attacker will be able to read the data even without knowing the passphrase.

If the automatically opening keystore is corrupted or lost, you must again enable automatic opening. The keystore.aks file will be recreated from keystore.ks at this time.

See

Refer to "pgx_rcvall" and "pgx_dmpall" in the Reference for information on the pgx_rcvall and pgx_dmpall commands.

Refer to "psql" under "Reference" in the PostgreSQL Documentation for information on the psql command.

Refer to "B.2 Transparent Data Encryption Control Functions" for information on the pgx_set_master_key function.

Refer to "5.6.3 Enabling Automatic Opening of the Keystore" for information on how to enable automatic opening of the keystore.

5.7 Backing Up and Restoring/Recovering the Database

Fujitsu Enterprise Postgres enables you to use the five backup and recovery methods described below. Regardless of the method you use, you must back up the keystore at the same time.

Note that you must store the database and the keystore on separate data storage media. Storing both on the same data storage medium risks the danger of the encrypted data being deciphered if the medium is stolen.

Backup and recovery using WebAdmin

- Backup

  WebAdmin backs up encrypted data.

  Back up the key store after backing up the database.

- Recovery

  Restore the keystore to its state at the time of database backup. Refer to "5.6.4 Backing Up and Recovering the Keystore" for details.

  Enable automatic opening of the keystore in accordance with the procedure described in "5.6.3 Enabling Automatic Opening of the Keystore". Then, use WebAdmin to recover the database.

Backup and recovery using the pgx_dmpall and pgx_rcvall commands

- Backup

  The pgx_dmpall command backs up the encrypted data.

  Back up the key store after backing up the database.

- Recovery

  Restore the keystore to its state at the time of the database backup.

  Configure automatic opening of the key store as necessary.

  If automatic opening of the keystore is not enabled, execute the pgx_rcvall command with the --keystore-passphrase option specified. This will display the prompt for the passphrase to be entered.

Example

- Back up the database and the keystore on March 1, 2022.

```
> pgx_dmpall -D D:\database\inst1
> copy C:\key\store\location\keystore.ks C:\keybackup\keystore_20220301.ks
```

  Specify the following in the pgx_dmpall command:

  - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

- Recover the database and the keystore from the backup taken on March 1, 2022.

```
> copy C:\keybackup\keystore_20220301.ks C:\key\store\location\keystore.ks
> pgx_keystore --enable-auto-open C:\key\store\location\keystore.ks (Execute only when enabling
automatic opening)
> pgx_rcvall -B E:\backup\inst1 -D D:\database\inst1 --keystore-passphrase
```

  Specify the following in the pgx_rcvall command:

  - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
  - Specify the backup data storage directory in the -B option.
  - The --keystore-passphrase option prompts you to enter the passphrase to open the keystore.

Dump and restore using SQL

- Backup

  The files output by the pg_dump and pg_dumpall commands are not encrypted. You should, therefore, encrypt the files using OpenSSL commands or other means before saving them, as described in "5.8 Importing and Exporting the Database" below.

  Back up the key store after backing up the database.

- Restore

  If the backup data has been encrypted using, for example Open SSL commands, decrypt that data.

  The data generated by the pg_dumpall command includes a specification to encrypt tablespaces by default. For this reason, the psql command encrypts tablespaces during restoration.

File system level backup and restore

- Backup

  Stop the instance and backup the data directory and the tablespace directory using the file copy command of the operating system. The files of encrypted tablespaces are backed up in the encrypted state.

  Back up the key store after performing the backup.

- Restore

  Restore the keystore to its state at the time of the database backup.

  Stop the instance and restore the data directory and the tablespace directory using the file copy command of the operating system.

Continuous archiving and point-in-time recovery

- Backup

  The pg_basebackup command backs up the encrypted data as is.

  Back up the key store after performing the backup.

- Recovery

  Restore the keystore to its state at the time of the database backup.

  Configure automatic opening of the key store as necessary.

  If automatic opening of the keystore is not enabled, refer to "5.3 Opening the Keystore" for information on starting an instance by specifying pg_ctl --keystore-passphrase start.

See

- Refer to "Reference" in the PostgreSQL Documentation for information on the following commands:
  - psql
  - pg_dump
  - pg_basebackup
- Refer to the Reference for information on the following commands:
  - pgx_rcvall
  - pgx_dmpall
  - pg_dumpall

If you have restored the keystore, repeat the process of enabling automatic opening of the keystore This ensures that the contents of the automatically opening keystore (keystore.aks) are identical to the contents of the restored keystore.

Refer to "5.6.3 Enabling Automatic Opening of the Keystore" for information on how to enable automatic opening of the keystore.

5.8 Importing and Exporting the Database

The files output by the COPY TO command are not encrypted. Therefore, when transferring files to other systems, you should encrypt files using OpenSSL commands, or use file transfer software that performs encrypted communication for Windows, to encrypt the data being transferred.

Use a safe method to delete obsolete plain text files.

You can use the following methods to safely delete files:

- fsutil command

Example

```
# Export the contents of the table my_table to a CSV file.
> psql -c "COPY my_table TO 'C:\WINDOWS\Temp\my_table.csv' (FORMAT CSV)" postgres
# Encrypt the exported file.
> C:\OpenSSL-Win32\bin\openssl enc -e -aes256 -in C:\WINDOWS\Temp\my_table.csv -out my_table.csv.enc
(The user is prompted to enter the passphrase to be used for encryption)

# Check the size of plain text files, and delete them after zero padding
> dir C:\WINDOWS\Temp\my_table.csv
> fsutil file setzerodata offset=0 length=7 C:\WINDOWS\Temp\my_table.csv
> del C:\WINDOWS\Temp\my_table.csv

# Decrypt the encrypted files on other systems.
> C:\OpenSSL-Win32\bin\openssl enc -d -aes256 -in my_table.csv.enc -out my_table.csv
(The user is prompted to enter the passphrase to be used for decryption)
```

If you use COPY FROM to import data to tables and indexes in an encrypted tablespace, the imported data is automatically encrypted before being stored.

5.9 Encrypting Existing Data

You cannot encrypt existing unencrypted tablespaces. In addition, you cannot change encrypted tablespaces so that they do not encrypt.

As an alternative, transfer the tables and indexes to other tablespaces. You can use the following SQL commands for this.

```
ALTER TABLE table_name SET TABLESPACE new_tablespace;
ALTER INDEX index_name SET TABLESPACE new_tablespace;
ALTER DATABASE database_name SET TABLESPACE new_tablespace;
```

See

Refer to "SQL Commands" under "Reference" in the PostgreSQL Documentation for information on SQL commands.

5.10 Operations in Cluster Systems

This section describes how to use transparent data encryption on cluster systems such as high-availability systems, streaming replication, and database multiplexing.

5.10.1 HA Clusters that do not Use Database Multiplexing

Take the following points into account when using transparent data encryption in an HA cluster environment that does not use database multiplexing.

Placement and automatic opening of the keystore file

There are two alternatives for placing the keystore file:

- Sharing the keystore file
- Placing a copy of the keystore file

Sharing the keystore file

This involves using the same keystore file on the primary server and the standby server.

As the standby server is not active while the primary server is running, this file would not be accessed simultaneously, and therefore, it can be shared.

To manage the keystore file in a more secure manner, place it on the key management server or the key management storage isolated in a secure location.

Enable the automatic opening of the keystore on both the primary and standby servers.

Placing a copy of the keystore file

This involves placing a copy of the primary server keystore file on the standby server.

You can do this if you cannot prepare a shared server or disk device that can be accessed from both the primary and standby servers.

However, if you change the master encryption key and the passphrase on the primary server, you must copy the keystore file to the standby server again.

To manage the keystore file in a more secure manner, prepare the key management server or the key management storage isolated in a secure location for both the primary and standby servers, and place the keystore files there.

Enable the automatic opening of the keystore on both the primary and standby servers. Note that copying the automatically opening keystore file (keystore.aks) to the standby server does not enable the automatic opening of the keystore.

5.10.2 Database Multiplexing Mode

Note the following when using transparent data encryption in environments that use streaming replication, or database multiplexing with streaming replication.

Placing the keystore file

Place a copy of the primary server keystore file on the standby server.

This is required as the keystore file cannot be shared, and both servers may need to access it simultaneously.

Point

To manage the keystore file in a more secure manner, place it on the key management server or the key management storage isolated in a secure location. A keystore used by both the primary and standby servers can be managed on the same key management server or key management storage.

However, create different directories for the keystores to be used by the primary server and the standby server. Then copy the keystore for the primary server to the directory used on the standby server.

Automatically opening the keystore

You must enable automatic opening of the keystore.

To do this, enable automatic opening of the keystore in all servers that make up database multiplexing. The settings for automatic opening of the keystore include information unique to each server, so simply copying the file does not enable it.

Changing the passphrase

Changes to the passphrase are reflected in all servers that make up database multiplexing, so no special operation is required.

Building and starting a standby server

Before using the pg_basebackup command or pgx_rcvall command to build a standby server, copy the keystore file from the primary server to the standby server. When using an automatically opening keystore, use the copied keystore file to enable automatic opening on the standby server.

Open the keystore each time you start the standby server. This step is necessary for decrypting and restoring encrypted WAL received from the primary server. To open the keystore, specify the --keystore-passphrase option in the pg_ctl command or pgx_rcvall command and enter the passphrase, or use an automatically opening keystore.

If specifying --keystore-passphrase in the pg_ct command, refer to "5.3 Opening the Keystore" for details.

Changing the master encryption key and the passphrase

Change the master encryption key and the passphrase on the primary server. You need not copy the keystore from the primary server to the standby server. You need not even restart the standby server or reopen the keystore. Changes to the master encryption key and the passphrase are reflected in the keystore on the standby server.

See

Refer to "pgx_rcvall " in the Reference for information on pgx_rcvall command.

Refer to "pg_basebackup" under "Reference" in the PostgreSQL Documentation for information on pg_basebackup command.

Refer to "High Availability, Load Balancing, and Replication" under "Server Administration" in the PostgreSQL Documentation for information on how to set up streaming replication.

5.11 Security-Related Notes

- Decrypted data is cached in the database server memory (shared buffer). As a result, unencrypted data is stored in a minidump, which is the process memory dump. You should, therefore, safely delete the memory dump. You can safely delete files by using the following command:
  - fsutil command
- Unencrypted data may be written from the database server memory to the operating system's swap area. To prevent leakage of information from the swap area, consider either disabling the use of swap area or encrypting the swap area using a full-disk encryption product.
- The content of the server log file is not encrypted. Therefore, in some cases the value of a constant specified in a SQL statement is output to the server log file. To prevent this, consider setting a parameter such as log_min_error_statement.

- When executing an SQL function that opens the keystore and modifies the master encryption key, ensure that the SQL statement containing the passphrase is not output to the server log file. To prevent this, consider setting a parameter such as log_min_error_statement. If you are executing this type of SQL function on a different computer from the database server, encrypt the communication between the client and the database server with SSL.
- Starting with FEP 10, logical replication is available, which allows non-backed up clusters to subscribe to databases where transparent data encryption is enabled. Logical replication does not need to have the same encryption strategy between publisher and subscriber.

  In this scenario, if the user wants to encrypt the subscribed copy of data as well, then it is the user's responsibility to create encryption policies to the subscribed databases. By default, published encrypted tablespace data will not be encrypted in the subscriber side.

5.12 Tips for Installing Built Applications

With transparent data encryption, you can easily encrypt all the data in an application without modifying the application. Database administrators install built applications in the following manner. However, this procedure stores data to the default tablespace, so take necessary action if processing differs from the original design.

1. (Normal procedure) Create an owner and a database for the built application.

```
CREATE USER crm_admin ...;
CREATE DATABASE crm_db ...;
```

2. (Procedure for encryption) Create an encrypted tablespace to store the data for the built application.

```
SET tablespace_encryption_algorithm = 'AES256';
CREATE TABLESPACE crm_tablespace LOCATION 'C:\crm\data';
```

3. (Procedure for encryption) Configure an encrypted tablespace as the default tablespace for the owner of the built application.

```
ALTER USER crm_admin SET default_tablespace = 'crm_tablespace';
ALTER USER crm_admin SET temp_tablespaces = 'crm_tablespace';
```

4. (Normal procedure) Install the built application. The application installer prompts you to enter the host name and the port number of the database server, the user name, and the database name. The installer uses the entered information to connect to the database server and execute the SQL script. For applications that do not have an installer, the database administrator must manually execute the SQL script.

Normally, the application's SQL script includes logic definition SQL statements, such as CREATE TABLE, CREATE INDEX, and GRANT or REVOKE, converted from the entity-relationship diagram. It does not include SQL statements that create databases, users, and tablespaces. Configuring the default tablespace of the users who will execute the SQL script deploys the objects generated by the SQL script to the tablespace.

Chapter 6 Data Masking

Data masking is a feature that can change the returned data for queries generated by applications, so that it can be referenced by users. For example, for a query of employee data, digits except the last four digits of an eight-digit employee number can be changed to "*" so that it can be used for reference.

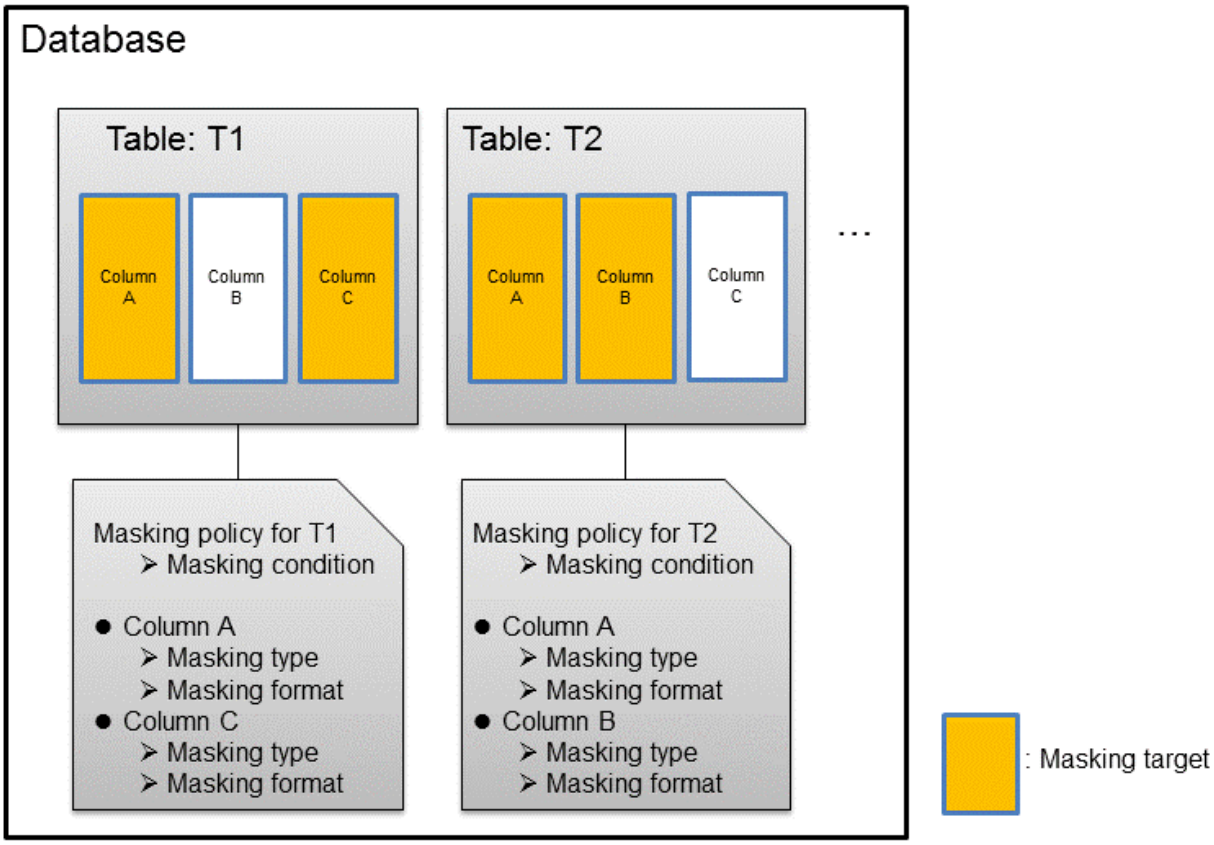
Note

When using this feature, it is recommended that the changed data be transferred to another medium for users to reference. This is because, if users directly access the database to extract the masked data, there is a possibility that they can deduce the original data by analyzing the masking policy or query result to the masking target column.

6.1 Masking Policy

Masking policy is a method of changing data under specific conditions when it is returned for a query from an application. One masking policy can be created per table. You can configure masking target, masking type, masking condition and masking format in a masking policy.

Figure 6.1 Masking policy



Note

When a masking policy is defined, the search performance for the corresponding table may deteriorate.

6.1.1 Masking Target

Masking target refers to a column to which a masking policy will be applied. When referring to a masking target or a function that includes a masking target, the execution result will be changed and obtained.

The following commands can change the execution result:

- SELECT
- COPY
- pg_dump
- pg_dumpall

- If a masking target is specified to INSERT...SELECT target columns, processing will be performed using data before change.
- If a masking target other than SELECT target columns is specified, processing will be performed using data before change.
- If a masking target is specified in a function where the data type will be converted, an error will occur.

6.1.2 Masking Type

Masking type is a method to change column data that is returned from queries. Specify the masking type in the function_type parameter. The following masking types can be specified and selected depending on the masking target data type.

Full masking

All the data in the specified column is changed. The changed value returned to the application that made the query varies depending on the column data type.
For example, 0 is used for a numeric type column and a space is used for a character type column.

Partial masking

The data in the specified column is partially changed.
For example, digits except the last four digits of an employee number can be changed to "*".

Regular expression masking

The data in the specified column is changed via a search that uses a regular expression.
For example, for strings such as email address that can have variable length, "*" can be used to change characters preceding "@" by using a regular expression. Regular expression masking can only be used for character type data.

- If multiple valid masking targets are specified for a function, the masking type for the left-most masking target will be applied.
  For example, if "SELECT GREATEST(c1, c2) FROM t1" is executed for numeric type masking target c1 and c2, the masking type for c1 will be applied.
- When masking the data that includes multibyte characters, do not specify partial masking for masking type. The result may not be as expected.

6.1.3 Masking Condition

Masking condition refers to the conditions configured to perform masking. Specify the masking condition in the expression parameter. Changed or actual data can be displayed for different users by defining masking condition. An expression that returns a boolean type result needs to be specified in masking condition and masking is performed only when TRUE is returned. Refer to "Value Expressions" in the PostgreSQL Documentation for information on the expressions that can be specified. Note that expressions that include a column cannot

be specified.
For example, when masking data only for "postgres" users, specify 'current_user = "postgres"' in the masking condition.

Information

Specify '1=1' so the masking condition is always evaluated to be TRUE and masking is performed all the time.

6.1.4 Masking Format

Masking format is a combination of change method and displayed characters when the masking condition is met. Masking format varies depending on the masking type. The following describes the masking format.

Full masking

With full masking, all characters are changed to values as determined by the database. Changed characters can be referenced in the pgx_confidential_values table. Also, replacement characters can be changed using the pgx_update_confidential_values system management function.

See

Refer to "6.3 Data Types for Masking" for information on the data types for which data masking can be performed.

Partial masking

With partial masking, data is changed according to the content in the function_parameters parameter. The method of specifying function_parameters varies depending on the data type.

Category	Method of specifying function_parameters
Numeric type	'*replacementCharacter*, *startPosition*, *endPosition*' - *replacementCharacter*: Specify the number to display. Specify a value from 0 to 9. - *startPosition*: Specify the start position of masking. Specify a positive integer. - *endPosition*: Specify the end position of masking. Specify a positive integer that is greater than *startPosition*. **Example** Specify as below to change the values from the 1st to 5th digits to 9. function_parameters := '9, 1, 5' In this example, if the original data is "123456789", it will be changed to "999996789".
Character type	*'inputFormat*, *outputFormat*, *replacementCharacter*, *startPosition*, *endPosition'* - *inputFormat*: Specify the current format of the data. Specify "V" for characters that will potentially be masked, and specify "F" for values such as spaces or hyphens that will not be masked. - *outputFormat*: Define the method to format the displayed data. Specify "V" for characters that will potentially be masked. Any character to be output can be specified for each character "F" in *inputFormat*. If you want to output a single quotation mark, specify two of them consecutively. - *replacementCharacter*: Specify any single character. If you want to output a single quotation mark, specify two of them consecutively. - *startPosition*: Specify the position of "V" as the start position of masking. For example, to specify the position of the 4th "V" from the left, specify 4. Specify a positive integer.

Category	Method of specifying function_parameters
	- *endPosition*: Specify the position of "V" as an end position of masking. When working out the end position, do not include positions of "F". For example, to specify the position of the 11th "V" from the left, specify 11. Specify a positive integer that is greater than *startPosition*. **Example** Specify as below to mask a telephone number other than the first three digits using *. function_parameters := 'VVVFVVVVFVVVV, VVV-VVVV-VVVV, *, 4, 11' In this example, if the original data is "012-3156-7890", it will be changed to "012-****-****".
Date/timestamp type	'MDYHMS' - M: Masks month. To mask month, enter the month from 1 to 12 after a lowercase letter m. Specify an uppercase letter M to not mask month. - D: Masks date. To mask date, enter the date from 1 to 31 after a lowercase letter d. If a value bigger than the last day of the month is entered, the last day of the month will be displayed. Specify an uppercase letter D to not mask date. - Y: Masks year. To mask year, enter the year from 1 to 9999 after a lowercase letter y. Specify an uppercase letter Y to not mask year. - H: Masks hour. To mask hour, enter the hour from 0 to 23 after a lowercase letter h. Specify an uppercase letter H to not mask hour. - M: Masks minute. To mask minute, enter the minute from 0 to 59 after a lowercase letter m. Specify an uppercase letter M to not mask minute. - S: Masks second. To mask second, enter the second from 0 to 59 after a lowercase letter s. Specify an uppercase letter S to not mask second. **Example** Specify as below to mask hour, minute, and second and display 00:00:00. function_parameters := 'MDYh0m0s0' In this example, if the original data is "2022-02-10 10:10:10", it will be changed to "2022-02-10 00:00:00".

See

- Refer to "B.3.2 pgx_create_confidential_policy" for information on function_parameters.
- Refer to "6.3 Data Types for Masking" for information on the data types for which masking can be performed.

Regular expression masking

With regular expression masking, data is changed according to the content of the regexp_pattern, regexp_replacement and regexp_flags parameters. For regexp_pattern, specify the search pattern using a regular expression. For regexp_replacement, specify the replacement character to use when data matches the search pattern. For regexp_flags, specify the regular expression flags.

Example

Specify as below to change all three characters starting from b to X.

regexp_pattern := 'b..'

regexp_replacement:= 'X'

regexp_flags := 'g'

In this example, if the original data is "foobarbaz", it will be changed to "fooXX".

See

- Refer to "POSIX Regular Expressions" in the PostgreSQL Documentation and check pattern, replacement, and flags for information on the values that can be specified for regexp_pattern, regexp_replacement, and regexp_flags.
- Refer to "6.3 Data Types for Masking" for information on the data types for which masking can be performed.

Note

- When column data type is character(*n*) or char(*n*) and if the string length after change exceeds n, the extra characters will be truncated and only characters up to the nth character will be displayed.
- When column data type is character varying(*n*) or varchar(*n*) and if the string length after change exceeds the length before the change, the extra characters will be truncated and only characters up to the length before change will be displayed.

6.2 Usage Method

Preparation

The following preparation is required to use this feature.

1. Set the postgresql.conf file parameters.

   Prepend "pgx_datamasking" to the shared_preload_libraries parameter.

2. Restart the instance.

3. Execute CREATE EXTENSION for the database that will use this feature.

   The target database is described as "postgres" here.

   Use the psql command to connect to the "postgres" database.

   #### Example

   ```
   postgres=# CREATE EXTENSION pgx_datamasking;
   CREATE EXTENSION
   ```

   #### Note

   You must always prepend "pgx_datamasking" to the "shared_preload_libraries" parameter.

   #### Information

   - Specify "false" for pgx_datamasking.enable to not use this feature. Data will not be masked even if a masking policy is configured. This feature becomes available again once "true" is specified for pgx_datamasking.enable. This setting can be made

by specifying a SET statement or specifying a parameter in the postgresql.conf file.
Example

```
postgres=# SET pgx_datamasking.enable=false;
```

- Hereafter, also perform this preparatory task for the "template1" database, so that this feature can be used by default when creating a new database.

Usage

To perform masking, a masking policy needs to be configured. The masking policy can be created, changed, confirmed, enabled, disabled or deleted during operation.
The procedures to perform these tasks are explained below with examples.

1. Creating a masking policy
2. Changing a masking policy
3. Confirming a masking policy
4. Enabling and disabling a masking policy
5. Deleting a masking policy

Note

Only database superusers can configure masking policies.

6.2.1 Creating a Masking Policy

An example of the operation on the server is shown below.

1. Create a masking policy
   Execute the pgx_create_confidential_policy system management function to create a masking policy.
   The following values are configured in this example.
   - Masking target: Numeric type c1
   - Masking type: FULL
   - Masking condition: '1=1'

```
postgres=# select pgx_create_confidential_policy(table_name := 't1', policy_name := 'p1',
expression := '1=1', column_name := 'c1', function_type := 'FULL');
 pgx_create_confidential_policy
--------------------------------
 t
(1 row)
```

2. Confirm the displayed data
   Confirm that the masking target data (column c1) has been correctly changed.

```
postgres=# select * from t1;
 c1 |      c2
----+--------------
  0 | 012-3456-7890
  0 | 012-3456-7891
  0 | 012-3456-7892
(3 row)
```

See

- Refer to "B.3.2 pgx_create_confidential_policy" for information on the pgx_create_confidential_policy system management function.

Note

- Only one masking policy can be created per table.
- All users can view the masking policy created, so do not grant the login privilege of the database where this feature is set to the users who refer to the changed data. Masking policies are defined in the "pgx_confidential_columns", "pgx_confidential_policies" and "pgx_confidential_values" tables.

6.2.2 Changing a Masking Policy

1. An example of the operation on the server is shown below.
2. Change a masking policy
   Execute the pgx_alter_confidential_policy system management function to change a masking policy.
   The following values are changed in this example.
   - Content of change: Add a masking target
   - Masking target: Character type c2
   - Masking type: PARTIAL
   - Masking condition: 'VVVFVVVVFVVVV, VVV-VVVV-VVVV, *, 4, 11'

```
postgres=# select pgx_alter_confidential_policy(table_name := 't1', policy_name := 'p1',
action := 'ADD_COLUMN', column_name := 'c2', function_type := 'PARTIAL', function_parameters :=
'VVVFVVVVFVVVV, VVV-VVVV-VVVV, *, 4, 11');
 pgx_alter_confidential_policy
-------------------------------
 t
(1 row)
```

3. Confirm the displayed data
   Confirm that the masking target data has been correctly changed.

```
postgres=# select * from t1;
 c1 |      c2
----+--------------
  0 | 012-****-****
  0 | 012-****-****
  0 | 012-****-****
(3 row)
```

See

- Refer to "B.3.1 pgx_alter_confidential_policy" for information on the pgx_alter_confidential_policy system management function.

6.2.3 Confirming a Masking Policy

An example of the operation on the server is shown below.

1. Confirm information about a masking target where a masking policy is set
   Refer to the pgx_confidential_columns table to confirm the masking target where the masking policy is set.

```
postgres=# select * from pgx_confidential_columns;
 schema_name | table_name | policy_name | column_name | function_type |
function_parameters           | regexp_pattern | regexp_replacement | regexp_flags |
column_description
-------------+------------+-------------+-------------+---------------
+---------------------------------------+----------------+--------------------+-------------
+--------------------
 public      | t1         | p1          | c1          | FULL
|                                       |                |                    |             |
```

```
 public      | t1         | p1          | c2          | PARTIAL       | VVVFVVVVFVVVV, VVV-VVVV-
VVVV, *, 4, 11 |                |                    |             |
(2 row)
```

2. Confirm information about the masking policy content
   Refer to pgx_confidential_policies to confirm the masking policy content.

```
postgres=# select * from pgx_confidential_policies;
 schema_name | table_name | policy_name | expression | enable | policy_description
-------------+------------+-------------+------------+--------+--------------------
 public      | t1         | p1          | 1=1        | t      |
(1 row)
```

See

- Refer to "D.1 pgx_confidential_columns" for information on the pgx_confidential_columns table.
- Refer to "D.2 pgx_confidential_policies" for information on the pgx_confidential_policies table.

6.2.4 Enabling and Disabling a Masking Policy

An example of the operation on the server is shown below.

1. Disable a masking policy
   Execute the pgx_enable_confidential_policy system management function to disable a masking policy.

```
postgres=# select pgx_enable_confidential_policy(table_name := 't1', policy_name := 'p1',
enable := 'f');
 pgx_enable_confidential_policy
--------------------------------
 t
(1 row)
```

2. Confirm the displayed data
   Confirm that the original data is displayed by disabling the masking policy.

```
postgres=# select * from t1;
 c1 |      c2
----+---------------
  1 | 012-3456-7890
  2 | 012-3456-7891
  3 | 012-3456-7892
(3 row)
```

3. Enable a masking policy
   Execute the pgx_enable_confidential_policy system management function to enable a masking policy.

```
postgres=# select pgx_enable_confidential_policy(table_name := 't1', policy_name := 'p1',
enable := 't');
 pgx_enable_confidential_policy
--------------------------------
 t
(1 row)
```

4. Confirm the displayed data
   Confirm that the masking target data has been correctly changed.

```
postgres=# select * from t1;
 c1 |      c2
----+---------------
  0 | 012-****-****
  0 | 012-****-****
```

```
  0 | 012-****-****
(3 row)
```

See

- Refer to "B.3.4 pgx_enable_confidential_policy" for information on the pgx_enable_confidential_policy system management function.

6.2.5 Deleting a Masking Policy

An example of the operation on the server is shown below.

1. Delete a masking policy
   Execute the pgx_drop_confidential_policy system management function to delete a masking policy.

```
postgres=# select pgx_drop_confidential_policy(table_name := 't1', policy_name := 'p1');
 pgx_drop_confidential_policy
------------------------------
 t
(1 row)
```

2. Confirm the displayed data
   Confirm that the original data is displayed by deleting the masking policy.

```
postgres=# select * from t1;
 c1 |      c2
----+--------------
  1 | 012-3456-7890
  2 | 012-3456-7891
  3 | 012-3456-7892
(3 row)
```

See

- Refer to "B.3.3 pgx_drop_confidential_policy" for information on the gx_drop_confidential_policy function.

6.3 Data Types for Masking

The data types for which data masking can be performed are shown below.

Category	Data type	Masking type		
		Full masking	Partial masking	Regular expression masking
Numeric type	smallint	Y	Y	N
	integer	Y	Y	N
	bigint	Y	Y	N
	decimal	Y	Y	N
	numeric	Y	Y	N
	float	Y	Y	N
	real	Y	Y	N
	double precision	Y	Y	N

Category	Data type	Masking type		
		Full masking	Partial masking	Regular expression masking
Character type	character varying(*n*)	Y	Y	Y
	varchar(*n*)	Y	Y	Y
	character(*n*)	Y	Y	Y
	char(*n*)	Y	Y	Y
Date/timestamp type	date	Y	Y	N
	timestamp	Y	Y	N

Note

Even if the data type can be masking, if the data is a special value (NaN, Infinity, -Infinity), it is not.

6.4 Security Notes

- Starting with FEP 10, logical replication is available, which allows non-backed up clusters to subscribe to databases where data masking policies are enabled. Logical replication allows publisher and subscriber databases to have their own or the same data masking policies.

  In this scenario, the user must disable data masking on the publisher database whenever a subscription is created. This ensures that subscribers are able to obtain the original data (initial copy) instead of the masked version. Then, it is the user's responsibility to set masking policies to each subscribed database.

- Take strong caution in publishing data masking's confidential tables (pgx_confidential_policies, pgx_confidential_columns, etc.) unless the user is publishing all tables of the database and wants to apply the same data masking's policies on the subscribed database for all of them.

  Otherwise, as these confidential tables contain the masking policies for all tables of the database, confidential policies of unpublished tables may be unintentionally published. Additionally, it is not possible to apply different data masking policies on the subscriber database.

Chapter 7 Periodic Operations

This chapter describes the operations that must be performed periodically when running daily database jobs.

7.1 Configuring and Monitoring the Log

Fujitsu Enterprise Postgres enables you to output database errors and warnings to a log file.

This information is useful for identifying if errors have occurred and the causes of those errors.

By default, this information is output to the event log. It is recommended that you configure Fujitsu Enterprise Postgres to collect logs from its log files (for example, log_destination) before operating Fujitsu Enterprise Postgres.

Periodically monitor the log files to check if any errors have occurred.

See

- Refer to "Error Reporting and Logging" under "Server Administration" in the PostgreSQL Documentation for information on logs.
- Refer to "Configuring Parameters" in the Installation and Setup Guide for Server for information on log settings when operating with WebAdmin.

7.2 Monitoring Disk Usage and Securing Free Space

When a database is used for an extended period, free space on the disk is continuously consumed and in some cases the disk space runs out. When this happens, database jobs may stop and no longer run.

You should, therefore, periodically monitor the usage of disk space, and delete obsolete files located in the disk.

Monitor the disk usage of the disk where the following directories are located:

- Data storage destination directory
- Transaction log storage destination (if the transaction log is stored in a different directory from the data storage destination directory)
- Backup data storage destination directory
- Tablespace storage destination directory

7.2.1 Monitoring Disk Usage

To check the disk usage, use the following operating system commands:

- fsutil volume diskfree command

You can even use SQL statements to check tables and indexes individually.

Refer to "Determining Disk Usage" under "Server Administration" in the PostgreSQL Documentation for information on this method.

Information

If you are using WebAdmin for operations, a warning is displayed when disk usage reaches 80%

7.2.2 Securing Free Disk Space

Secure free disk space by using the following operating system commands to delete unnecessary files, other than the database, from the same disk unit.

- del command

You can also secure disk space by performing the following tasks periodically:

- To secure space on the data storage destination disk:

  Execute the REINDEX statement. Refer to "7.5 Reorganizing Indexes" for details.

- To secure space on the backup data storage destination disk:

  Execute backup using WebAdmin or the pgx_dmpall command.

7.3 Automatically Closing Connections

If an application stops responding and abnormally terminates for any reason, the connection from the application may remain active on the database server. If this situation continues for an extended period, other applications attempting to connect to the database server may encounter an error, or an error indicating that the tables are unavailable may occur.

It is, therefore, recommended that idle connections be closed automatically at regular intervals.

Set the following parameters in the postgresql.conf file to indicate the time permitted to elapse before a connection is closed.

Parameter name	Setting	Description
tcp_keepalives_idle	Time until keepalive is sent (seconds) If 0, the default value of the system is used.	Sends keepalive to an idle connection at the specified interval in seconds It is recommended to specify 30 seconds.
tcp_keepalives_interval	keepalive send interval (seconds) If 0, the default value of the system is used.	Sends keepalive at the specified interval It is recommended to specify 6 seconds.

Note

The maximum number of connections allowed is 125, unless the desktop heap setting is changed.

See

Refer to "Connection Settings" under "Server Administration" in the PostgreSQL Documentation for information on the parameters.

7.4 Monitoring the Connection State of an Application

Fujitsu Enterprise Postgres does not immediately delete the updated or deleted data. If the VACUUM determines there are no transactions that reference the database, Fujitsu Enterprise Postgres collects obsolete data.

However, obsolete data is not collected if there are connections that have remained active for an extended period or connections occupying resources. In this case the database may expand, causing performance degradation.

See

Refer to "Routine Vacuuming" under "Server Administration" in the PostgreSQL Documentation for information on the VACUUM command.

In such cases, you can minimize performance degradation of the database by monitoring problematic connections.

The following methods are supported for monitoring connections that have been in the waiting status for an extended period:

- 7.4.1 Using the View (pg_stat_activity)
- 7.4.2 Using pgAdmin

7.4.1 Using the View (pg_stat_activity)

Use the view (pg_stat_activity) to identify and monitor connections where the client has been in the waiting status for an extended period.

Example

The example below shows connections where the client has been in the waiting status for at least 60 minutes.

However, when considering continued compatibility of applications, do not reference system catalogs directly in the following SQL statements.

```
postgres=# select * from pg_stat_activity where backend_type = 'client backend' and state='idle in
transaction' and current_timestamp > cast(query_start + interval '60 minutes' as timestamp);
-[ RECORD 1 ]----+------------------------------
datid            | 13003
datname          | db01
pid              | 4638
leader_pid       |
usesysid         | 10
usename          | fsep
application_name | apl01
client_addr      | 192.33.44.15
client_hostname  |
client_port      | 27500
backend_start    | 2022-02-24 09:09:21.730641+09
xact_start       | 2022-02-24 09:09:23.858727+09
query_start      | 2022-02-24 09:09:23.858727+09
state_change     | 2022-02-24 09:09:23.858834+09
wait_event_type  | Client
wait_event       | ClientRead
state            | idle in transaction
backend_xid      |
backend_xmin     |
query_id         |
query            | begin;
backend_type     | client backend
```

See

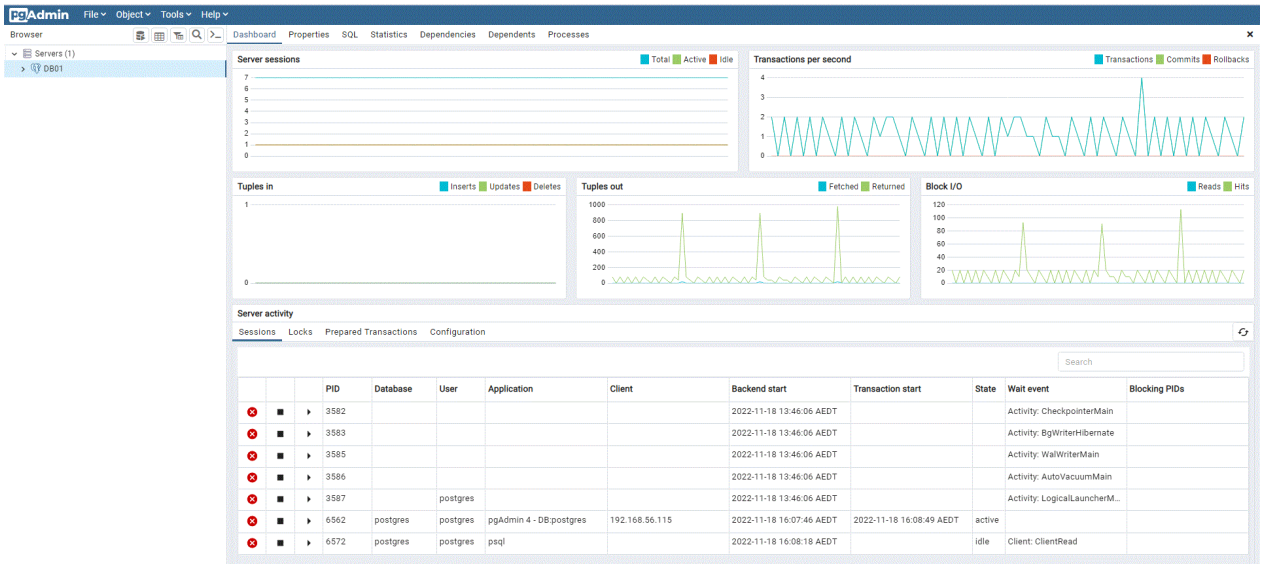
- Refer to "Notes on Application Compatibility" in the Application Development Guide for information on maintaining application compatibility.
- Refer to "The Statistics Collector" under "Server Administration" in the PostgreSQL Documentation for information on pg_stat_activity.

7.4.2 Using pgAdmin

This section describes the procedure for monitoring connections using [Server Status] in pgAdmin.

1. In the [Browser] pane, click the database server for monitoring.

2. In the [Dashboard] tab, identify client connections that have been in the waiting state for an extended period.



7.5 Reorganizing Indexes

Normally, a database defines indexes in tables, but if data is frequently updated, indexes can no longer use free space in the disk efficiently. This situation can also cause a gradual decline in database access performance.

To rearrange used space on the disk and prevent the database access performance from declining, it is recommended that you periodically execute the REINDEX command to reorganize indexes.

Check the disk usage of the data storage destination using the method described in "7.2 Monitoring Disk Usage and Securing Free Space".

Note

Because the REINDEX command retrieves the exclusive lock for an index being processed and locks writing of tables that are the source of the index, other processes that access these may stop while waiting to be locked.

Therefore, it is necessary to consider measures such as executing the command after the task is completed.

See

Refer to "Routine Reindexing" under "Server Administration" in the PostgreSQL Documentation for information on reorganizing indexes by periodically executing the REINDEX command.

Point

Typically, reorganize indexes once a month at a suitable time such as when conducting database maintenance. Use SQL statements to check index usage. If this usage is increasing on a daily basis, adjust the frequency of recreating the index as compared to the free disk space.

The following example shows the SQL statements and the output.

However, when considering continued compatibility of applications, do not reference system catalogs and functions directly in the following SQL statements. Refer to "Notes on Application Compatibility" in the Application Development Guide for details.

[SQL statements]

```
SELECT
  nspname AS schema_name,
  relname AS index_name,
```

```
  round(100 * pg_relation_size(indexrelid) / pg_relation_size(indrelid)) / 100 AS index_ratio,
  pg_size_pretty(pg_relation_size(indexrelid)) AS index_size,
  pg_size_pretty(pg_relation_size(indrelid)) AS table_size
FROM pg_index I
  LEFT JOIN pg_class C ON (C.oid = I.indexrelid)
  LEFT JOIN pg_namespace N ON (N.oid = C.relnamespace)
WHERE
  C.relkind = 'i' AND
  pg_relation_size(indrelid) > 0
ORDER BY pg_relation_size(indexrelid) DESC, index_ratio DESC;
```

[Output]

```
 schema_name |            index_name            | index_ratio | index_size | table_size
-------------+----------------------------------+-------------+------------+------------
 public      | pgbench_accounts_pkey            |        0.16 | 2208 KB    | 13 MB
 pg_catalog  | pg_depend_depender_index         |         0.6 | 224 KB     | 368 KB
 pg_catalog  | pg_depend_reference_index        |        0.58 | 216 KB     | 368 KB
...
```

See

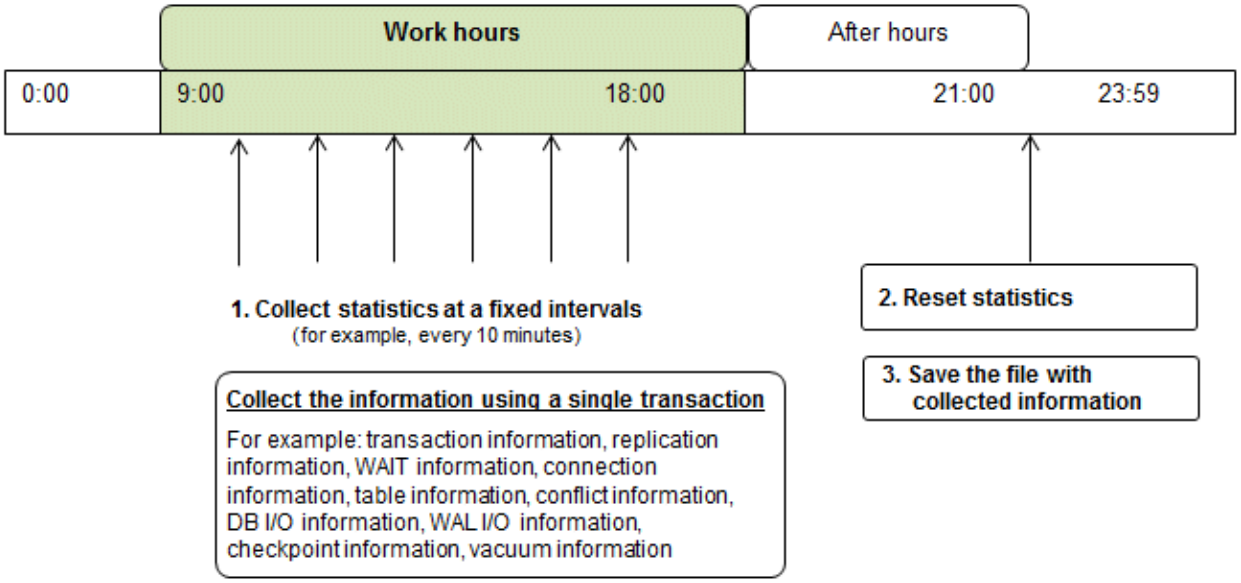
Refer to "Notes on Application Compatibility" in the Application Development Guide for information on maintaining application compatibility.

7.6 Monitoring Database Activity

Fujitsu Enterprise Postgres enables you to collect information related to database activity. By monitoring this information, you can check changes in the database status.

This information includes wait information for resources such as internal locks, and is useful for detecting performance bottlenecks. Furthermore, you should collect this information in case you need to request Fujitsu technical support for an investigation.

Figure 7.1 Overview of information collection



1. Collect statistics at fixed intervals during work hours.

   Accumulate the collected information into a file.

Wherever possible, collect data from the various statistics views using a single transaction, because it enables you to take a snapshot of system performance at a given moment.

Refer to "7.6.1 Information that can be Collected" for information on the system views that can be collected.

2. Reset statistics after work hours, that is, after jobs have finished.

   Refer to "7.6.3 Information Reset" for information on how to reset statistics.

3. Save the file with collected information.

   Keep the file with collected information for at least two days, in order to check daily changes in performance and to ensure that the information is not deleted until you have sent a query to Fujitsu technical support.

Where jobs run 24 hours a day, reset statistics and save the file with collected information when the workload is low, for example, at night.

Statistics cumulatively add the daily database value, so if you do not reset them, the values will exceed the upper limit, and therefore will not provide accurate information.

The subsections below explain the following:

- Information that can be collected
- Collection configuration
- Information reset

7.6.1 Information that can be Collected

Information that can be collected is categorized into the following types:

- Information common to PostgreSQL
- Information added by Fujitsu Enterprise Postgres

Information common to PostgreSQL

Refer to "Monitoring Database Activity" under "Server Administration" in the PostgreSQL Documentation for information on information common to PostgreSQL.

Information added by Fujitsu Enterprise Postgres

You can collect the following information added by Fujitsu Enterprise Postgres.

Table 7.1 Information added by Fujitsu Enterprise Postgres

View name	Description
pgx_stat_lwlock	Displays statistic related to lightweight lock, with each type of content displayed on a separate line. This information helps to detect bottlenecks. Refer to "C.2 pgx_stat_lwlock" for details.
pgx_stat_latch	Displays statistics related latches, with each type of wait information within Fujitsu Enterprise Postgres displayed on a separate line. This information helps to detect bottlenecks. Refer to "C.3 pgx_stat_latch" for details.
pgx_stat_walwriter	Displays statistics related to WAL writing, in a single line. Refer to "C.4 pgx_stat_walwriter" for details.

View name	Description
pgx_stat_sql	Displays statistics related to SQL statement executions, with each type of SQL statement displayed on a separate line. Refer to "C.5 pgx_stat_sql" for details.
pgx_stat_gmc	Displays statistics related to Global Meta Cache hit ration and used memory size. Refer to "C.6 pgx_stat_gmc" for detail. Also refer to Chapter 12 Global Meta Cache" for information on the Global Meta Cache.

7.6.2 Collection Configuration

The procedure for configuring collection depends on the information content.

- Information common to PostgreSQL
- Information added by Fujitsu Enterprise Postgres

Information common to PostgreSQL

Refer to "The Statistics Collector" in "Monitoring Database Activity" under "Server Administration" in the PostgreSQL Documentation for information on information common to PostgreSQL.

Information added by Fujitsu Enterprise Postgres

Information added by Fujitsu Enterprise Postgres is collected by default.

To enable or disable information collection, change the configuration parameters in postgresql.conf. The following table lists the views for which you can enable or disable information collection, and the configuration parameters.

View name	Parameter
pgx_stat_lwlock pgx_stat_latch	track_waits (*1)
pgx_stat_sql	track_sql
pgx_stat_gmc	track_gmc

Remarks: You cannot change the collection status for pgx_stat_walwriter.

*1: When executing the SQL statement with EXPLAIN ANALYZE, processing time may increase because of this information collection. It is recommended to set this parameter to "off" when executing EXPLAIN ANALYZE to check the processing time.

Refer to "Appendix A Parameters" for information on the parameters.

7.6.3 Information Reset

This section describes how to reset information.

Information added by Fujitsu Enterprise Postgres

You can reset information added by Fujitsu Enterprise Postgres by using the pg_stat_reset_shared function in the same way as for information common to PostgreSQL.

Configure the following parameters in the pg_stat_reset_shared function:

Function	Type of return value	Description
pg_stat_reset_shared(text)	void	Reset some cluster-wide statistics counters to zero, depending on the argument (requires superuser privileges). Calling pg_stat_reset_shared('lwlock') will zero all counters shown in pgx_stat_lwlock. Similarly, in the following cases, all values of the pertinent statistics counter are reset: - If pg_stat_reset_shared('latch') is called: All values displayed in pgx_stat_latch - If pg_stat_reset_shared('walwriter') is called: All values displayed in pgx_stat_walwriter - If pg_stat_reset_shared('sql') is called: All values displayed in pgx_stat_sql - If pg_stat_reset_shared('gmc') is called: All values except size column in pgx_stat_gmc

See

Refer to "Statistics Functions" in "Monitoring Database Activity" under "Server Administration" in the PostgreSQL Documentation for information on other parameters of the pg_stat_reset_shared function.

Chapter 8 Streaming Replication Using WebAdmin

This chapter describes how to create a streaming replication cluster using WebAdmin.

Streaming replication allows the creation of one or more standby instances, which connect to the master instances and replicate the data using WAL records. The standby instance can be used for read-only operations.

WebAdmin can be used to create a streaming replication cluster. WebAdmin allows the creation of a cluster in the following configurations:

- Master-Standby Configuration: This configuration creates a master and standby instance together.
- Standby Only Configuration: This configuration creates a standby instance from an already existing instance.

Point

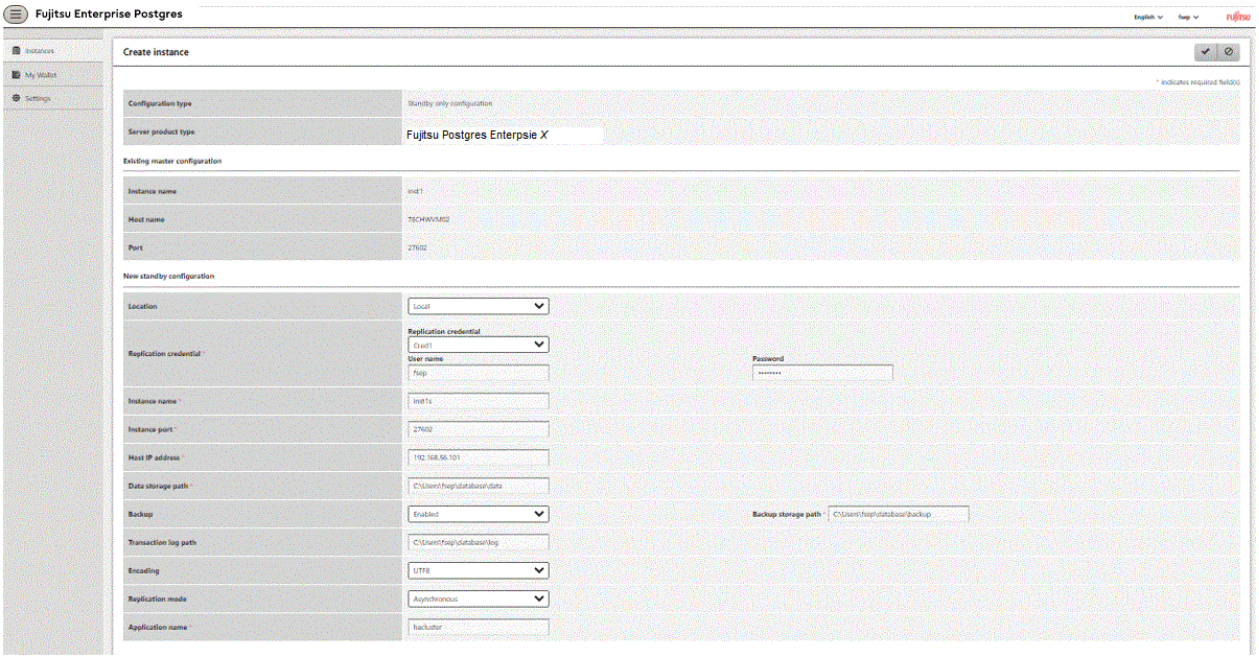
- A standby instance can be created from a standalone instance, a master instance, or even from another standby instance.
- If a streaming replication cluster is created using WebAdmin, the network with the host name (or IP address) specified in [Host name] will be used across sessions of WebAdmin, and also used as the log transfer network.
- To use a network other than the job network as the log transfer network, specify the host name other than the job network one in [Host name].

8.1 Creating a Standby Instance

Follow the procedure below to create a standby instance.

1. In the [Instances] tab, select the instance from which a standby instance is to be created.
2. Click .
3. Enter the information for the standby instance to be created. In the example below, a standby instance is created from instance "inst1".

   The instance name, host address and port of the selected instance are already displayed for easy reference.



   Enter the following items:

   - [Location]: Whether to create the instance in the server that the current user is logged in to, or in a remote server. The default is "Local", which will create the instance in the server machine where WebAdmin is currently running.

- [Replication credential]: The user name and password required for the standby instance to connect to the master instance. The user name and password can be entered or selected from the Wallet. Refer to "Appendix G WebAdmin Wallet" for information on creating wallet entries.
- [Instance name]: Name of the standby database instance to create.

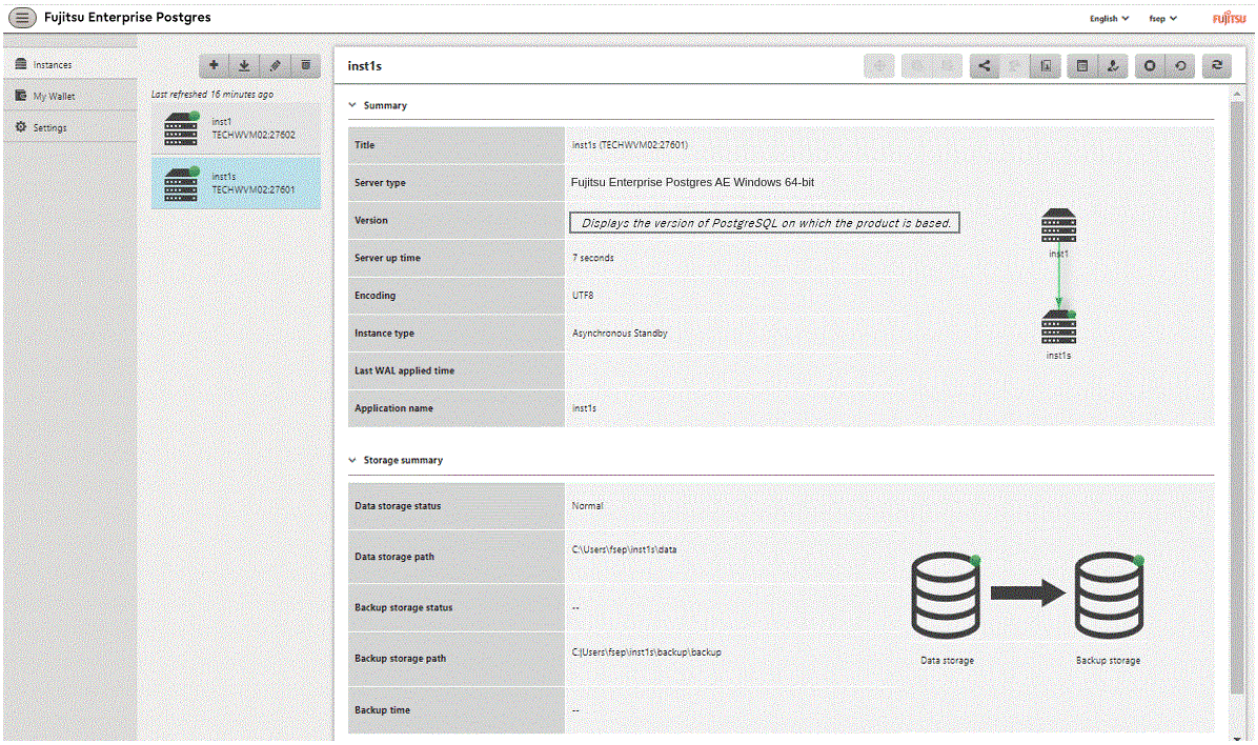
  The name must meet the conditions below:

  - Maximum of 16 characters
  - The first character must be an ASCII alphabetic character
  - The other characters must be ASCII alphanumeric characters
- [Instance port]: Port number of the standby database instance.
- [Host IP address]: The IP address of the server machine where the standby instance is to be created. This information is needed to configure the standby instance to be connected to the master.
- [Data storage path]: Directory where the database data will be stored
- [Backup storage path]: Directory where the database backup will be stored
- [Transaction log path]: Directory where the transaction log will be stored
- [Encoding]: Database encoding system
- [Replication mode]: Replication mode of the standby instance to be created ("Asynchronous" or "Synchronous")
- [Application name]: The reference name of the standby instance used to identify it to the master instance.

  The name must meet the conditions below:

  - Maximum of 16 characters
  - The first character must be an ASCII alphabetic character
  - The other characters must be ASCII alphanumeric characters

4. Click ✔ to create the standby instance.
5. Once the standby instance is created successfully, select standby instance in the [Instances] tab. The following page will be displayed:



Note

- Backups are not possible for standby instances in WebAdmin. As a result, and are disabled and no value is shown for [Backup storage status] and [Backup time].
- If using WebAdmin to manage Mirroring Controller, the message below may be output to the server log or system log in the standby instance. No action is required, as the instance is running normally.

```
ERROR:  pgx_rcvall failed
ERROR:  pgx_rcvall: backup of the database has not yet been performed, or an incorrect backup
storage directory was specified
```

- Replication credential (user name and password) should not contain hazardous characters. Refer to "Appendix H WebAdmin Disallow User Inputs Containing Hazardous Characters".

8.2 Promoting a Standby Instance

Streaming replication between a master and standby instance can be discontinued using WebAdmin.

Follow the procedure below to promote a standby instance to a standalone instance, thereby discontinuing the streaming replication.

1. In the [Instances] tab, select the standby instance that needs to be promoted.
2. Click .
3. Click [Yes] from the confirmation dialog box.

The standby instance will be promoted and will become a standalone instance, which is not part of a streaming replication cluster.

Once the standby instance is promoted to become a standalone instance, the backup storage status will be "Error". This is because no backups are available when the instance is newly promoted to a standalone instance. The status will be reset if a new backup is performed by clicking [Solution] or .

8.3 Converting an Asynchronous Replication to Synchronous

Streaming replication between a master and standby instance can be configured to be in Asynchronous or Synchronous mode. This mode can be changed even after the standby instance was successfully created.

Follow the procedure below to convert an Asynchronous standby instance to Synchronous.

1. In the [Instances] tab, select the master instance of the relevant cluster.
2. Click .
3. In the [Streaming replication] section, edit the value for [Synchronous standby names].
   - Add the "Application name" of the standby instance you want to be in Synchronous mode.
4. Click .
5. Select the master instance and click .
6. Select the standby instance. [Instance type] will now show the updated status.

Note

- Converting an Asynchronous standby instance to Synchronous can cause the master instance to queue the incoming transactions until the standby instance is ready. For this reason, it is recommended that this operation be performed during a scheduled maintenance period.
- When adding a synchronous standby instance, Fujitsu Enterprise Postgres will only keep the first entry in [Synchronous standby names] in synchronous state.

- To learn more about the differences between synchronous and asynchronous standby modes and their behavior, refer to "Streaming Replication" in "High Availability, Load Balancing, and Replication" in the PostgreSQL Documentation.

8.4 Converting a Synchronous Replication to Asynchronous

Streaming replication between a master and standby instance can be configured to be in Asynchronous or Synchronous Mode. This mode can be changed even after the standby instance was successfully created.

Follow the procedure below to convert a Synchronous standby instance to Asynchronous.

1. In the [Instances] tab, select the master instance of the relevant cluster.
2. Click .
3. In the [Streaming replication] section, edit the value for [Synchronous standby names].
   - Remove the "Application name" of the standby instance you want to be in Asynchronous mode.
4. Click .
5. Select the master instance and click .
6. Select the standby instance. [Instance type] will now show the updated status.

Note

To learn more about the differences between synchronous and asynchronous standby modes and their behavior, refer to "Streaming Replication" in "High Availability, Load Balancing, and Replication" in the PostgreSQL Documentation.

8.5 Joining a Replication Cluster

WebAdmin facilitates the joining of an old master of the cluster as a standby node.

1. In the [Instances] tab, select the remote instance (from where the new cluster node will stream WAL entries), and then click .
2. Configure the node to accept streaming requests from the new node.
3. In the [Instances] tab, select the new standby instance (which needs to be connected to the cluster), and then click .
4. Set [Replication host name] to the remote instance.
5. Enter [Replication credential].

   Specify the user name and password required for the standby instance to connect to the remote instance. The user name and password can be entered or selected from the Wallet. Refer to "Appendix G WebAdmin Wallet" for information on creating wallet entries. Replication credential (user name and password) should not contain hazardous characters. Refer to "Appendix H WebAdmin Disallow User Inputs Containing Hazardous Characters".
6. Enter [Host IP address].

   Specify the IP address of the node where the standby instance was created.
7. Click to open the [Join replication cluster] dialog box.

   For Fujitsu Enterprise Postgres 13 and earlier instances, select [Restart later] or [Restart now], and then click [Yes] to set up the standby instance.
   For Fujitsu Enterprise Postgres 14 or later, click [Yes] to set up the standby instance.
8. Upon successful completion, the confirmation dialog box will be displayed.
9. Click [Close] to return to the instance details window.

The instance will become a standby instance, and will be part of the streaming replication cluster. The replication diagram will display the relationship between the standby instance and the remote instance. The user can change the replication relationship of the remote instance from asynchronous to synchronous (and vice versa) using the [Configuration] window.

Chapter 9 Installing and Operating the In-memory Feature

The in-memory feature enables fast aggregation using Vertical Clustered Index (VCI) and memory-resident feature.

VCI has a data structure suitable for aggregation, and features parallel scan and disk compression, which enable faster aggregation through reduced disk I/O.

The memory-resident feature reduces disk I/O that occurs during aggregation. It consists of the preload feature that reads VCI data to memory in advance, and the stable buffer feature that suppresses VCI data eviction from memory. The stable buffer feature secures the proportion specified by parameter in the shared memory for VCI.

This chapter describes how to install and operate the in-memory feature.

Note

This feature can only be used in Advanced Edition.

9.1 Installing Vertical Clustered Index (VCI)

This section describes the installation of VCI.

1. Evaluating whether to Install VCI
2. Estimating Resources
3. Setting up

9.1.1 Evaluating whether to Install VCI

VCI uses available resources within the server to increase scan performance.

It speeds up processing in many situations, and can be more effective in the following situations:

- Single table processing
- Processing that handles many rows in the table
- Processing that handles some columns in the table
- Processing that performs very heavy aggregation such as simultaneous sum and average aggregation

VCI will not be used in the following cases, so it is necessary to determine its effectiveness in advance:

- The data type of the target table or column contains VCI restrictions.
- The SQL statement does not meet the VCI operating conditions
- VCI is determined to be slower based on cost estimation

Note

If performing operations that use VCI, the full_page_writes parameter setting in postgresql.conf must be enabled (on). For this reason, if this parameter is disabled (off), operations that use VCI return an error. In addition, to perform operations for tables that do not create a VCI when the full_page_writes parameter setting is temporarily disabled (off), do not create a VCI or perform operations to tables that created a VCI during that time.

See

- Refer to "9.1.4 Data that can Use VCI" for information on VCI restrictions.

- Refer to "Scan Using a Vertical Clustered Index (VCI)" - "Operating Conditions" in the Application Development Guide for information on VCI operating conditions.

9.1.2 Estimating Resources

Estimate resources before setting up VCI.

Select the aggregation that you want to speed up and identify the required column data. The additional resources below are required according to the number of columns.

- Memory

  Secure additional capacity required for the disk space for the column for which VCI is to be created.

- Disk

  Secure additional disks based on the disk space required for the column for which VCI is to be created, as VCI stores column data as well as existing table data on the disk. It is recommended to provide a separate disk in addition to the existing one, and specify it as the tablespace to avoid impact on any other jobs caused by I/O.

Information

The operations on VCI can continue even if the memory configured for VCI is insufficient by using VCI data on the disk.

See

Refer to "Estimating Memory Requirements" and "Estimating Database Disk Space Requirements" in the Installation and Setup Guide for Server for information on how to estimate required memory and disk space.

9.1.3 Setting up

This section describes how to set up VCI.

Setup flow

1. Setting Parameters
2. Installing the Extensions
3. Creating VCI
4. Confirming that VCI has been Created

9.1.3.1 Setting Parameters

Edit postgresql.conf to set the required parameters for VCI. After that, start or restart the instance.

The following table lists the parameters that need or are recommended to be configured in advance:

Parameter name	Setting value	Description	Required
shared_preload_libraries	Literal 'vci, pg_prewarm'	VCI and shared library to be preloaded at server start.	Y
session_preload_libraries	Literal 'vci, pg_prewarm'	VCI and shared library to be preloaded at connection start.	Y
reserve_buffer_ratio	Percentage of shared memory to be used for stable buffer table	Proportion of shared memory to be used for a stable buffer table.	N

Parameter name	Setting value	Description	Required
vci.control_max_workers	Number of background workers that manage VCI	Number of background workers that manage VCI. Add this value to max_worker_processes.	N
vci.max_parallel_degree	Maximum number of background workers used for parallel scan	Maximum number of background workers used for parallel scan. Add this value to max_worker_processes.	N
vci.smc_directory	Directory name in which a temporary file is created as the dynamic shared memory	Directory name in which a temporary file is created as the dynamic shared memory during a scan using a VCI.	N

Example

```
shared_preload_libraries =  'vci, pg_prewarm'
session_preload_libraries =  'vci, pg_prewarm'
reserve_buffer_ratio = 20
vci.control_max_workers = 8
vci.max_parallel_degree = 4
max_worker_processes = 18 # Example: If the initial value was 6, 6 + 8 + 4 = 18
vci.smc_directory = 'E:\\vci\\work'
```

Note

If performing operations that use VCI, do not delete the shared library names specified to "shared_preload_libraries" and "session_preload_libraries". If operations that access VCI are performed after the VCI was defined, unexpected behavior may occur.

See

- Refer to "Appendix A Parameters" for information on all parameters for VCI. Refer also to default value for each parameter and details such as specification range in the same chapter. Refer to "Server Configuration" under "Server Administration" in the PostgreSQL documentation for information on shared_preload_libraries, session_preload_libraries, and max_worker_processes.

9.1.3.2 Installing the Extensions

Execute CREATE EXTENSION to install the VCI and pg_prewarm extensions. Both extensions need to be installed for each database.

- Installing VCI

```
db01=# CREATE EXTENSION vci;
```

- Installing pg_prewarm

```
db01=# CREATE EXTENSION pg_prewarm;
```

Note

- Only superusers can install VCI extensions.
- VCI extensions can only be installed in public schema.
- Some operations cannot be performed for VCI extensions. Refer to "9.2.1 Commands that cannot be Used for VCI" for details.

9.1.3.3 Creating a VCI

Execute the CREATE INDEX statement with the "USING vci" clause to create a VCI for the desired columns and the "WITH (stable_buffer=true)" clause to enable the stable buffer feature.

To use a separate disk for the VCI, specify the TABLESPACE clause.

```
db01=# CREATE INDEX idx_vci ON table01 USING vci (col01, col02) WITH (stable_buffer=true);
```

Note

- Some table types cannot be specified on the ON clause of CREATE INDEX. Refer to "9.1.4.1 Relation Types" for details.
- Some data types cannot be specified on the column specification of CREATE INDEX. Refer to "9.1.4.2 Data Types" for details.
- Some operations cannot be performed for VCI. Refer to "9.2.1 Commands that cannot be Used for VCI" for details.
- The same column cannot be specified more than once on the column specification of CREATE INDEX.
- VCI cannot be created for table columns that belong to the template database.
- CREATE INDEX creates multiple views named vci_*10digitRelOid_5digitRelAttr_1charRelType* alongside VCI itself. These are called VCI internal relations. Do not update or delete them as they are used for VCI aggregation.
- All data for the specified column will be replaced in columnar format when VCI is created, so executing CREATE INDEX on an existing table with data inserted takes more time compared with a general index (B-tree). Jobs can continue while CREATE INDEX is running.
- When CREATE INDEX USING VCI is invoked on a partitioned table, the default behavior is to recurse to all partitions to ensure they all have matching indexes. Each partition is first checked to determine whether an equivalent index already exists, and if so, that index will become attached as a partition index to the index being created, which will become its parent index. If no matching index exists, a new index will be created and automatically attached; the name of the new index in each partition will be determined as if no index name had been specified in the command. If the ONLY option is specified, no recursion is done, and the index is marked invalid. (ALTER INDEX ... ATTACH PARTITION marks the index valid, once all partitions acquire matching indexes.) Note, however, that any partition that is created in the future using CREATE TABLE ... PARTITION OF will automatically have a matching index, regardless of whether ONLY is specified.
- Parallel index build is not supported on VCI indexes.

9.1.3.4 Confirming that the VCI has been Created

Execute the SELECT statement to reference the pg_indexes catalog, and confirm that the VCI was created for the target columns.

Example

```
db01=# SELECT indexdef FROM pg_indexes WHERE indexdef LIKE '%vci%';
                          indexdef
-----------------------------------------------------
CREATE INDEX idx_vci ON table01 USING vci (col01, col02)
(1 row)
```

9.1.4 Data that can Use VCI

This section describes on which relation types and for which data types VCIs can be created.

9.1.4.1 Relation Types

VCIs cannot be created on some relation types.

The ON clause of CREATE INDEX described in "9.1.3.3 Creating a VCI" cannot specify relations on which VCIs cannot be created.

- Relations on which VCIs can be created
  - Normal tables
  - UNLOGGED TABLEs
- Relations on which VCIs cannot be created
  - Materialized views
  - Temporary tables
  - Views
  - Temporary views
  - Foreign tables

9.1.4.2 Data Types

VCIs cannot be created for some data types.

The column specification of CREATE INDEX described in "9.1.3.3 Creating a VCI" cannot specify a column with data type on which VCIs cannot be created.

Category	Data type	Supported by VCI?
Numeric	smallint	Y
	integer	Y
	bigint	Y
	decimal	Y
	numeric	Y
	real	Y
	double precision	Y
	serial	Y
	bigserial	Y
Monetary	money	Y
Character	varchar(*n*)	Y
	char(*n*)	Y
	nchar	Y
	nvarchar(*n*)	Y
	text	Y
Binary	bytea	Y
Date/time	timestamp	Y
	timestamp with time zone	Y
	date	Y
	time	Y
	time with time zone	Y
	interval	Y
Boolean	boolean	Y
Geometric	point	N

Category	Data type	Supported by VCI?
	line	N
	lseg	N
	box	N
	path	N
	polygon	N
	circle	N
Network address	cidr	N
	inet	N
	macaddr	N
	macaddr8	N
Bit string	bit(*n*)	Y
	bit varying(*n*)	Y
Text search	tsvector	N
	tsquery	N
UUID	uuid	Y
XML	xml	N
JSON	json	N
	jsonb	N
Range	int4range	N
	int8range	N
	numrange	N
	tsrange	N
	tstzrange	N
	daterange	N
Object identifier	oid	N
	regproc	N
	regprocedure	N
	regoper	N
	regoperator	N
	regclass	N
	regtype	N
	regconfig	N
	regdictionary	N
pg_lsn type	pg_lsn	N
Array type	-	N
User-defined type (Basic type, enumerated type, composite type, and range type)	-	N

9.2 Operating VCI

This section describes how to operate VCI.

9.2.1 Commands that cannot be Used for VCI

Some operations cannot be performed for VCI extensions and VCI itself.

This section describes SQL commands that cannot be executed for the VCI extensions and VCI itself, and client application commands.

SQL commands

- Operations that cannot be performed for the VCI extension

Command	Subcommand	Description
ALTER EXTENSION	UPDATE	The VCI extension cannot be specified. This operation is not required for VCI.
	SET SCHEMA	
	ADD	
	DROP	
CREATE EXTENSION	SCHEMA	The subcommands on the left cannot be performed if the VCI extension is specified. This operation is not required for VCI.

- Operations that cannot be performed on relations containing a VCI

Command	Subcommand	Description
ALTER INDEX	SET	The subcommands on the left cannot be performed if a VCI is specified. If the operation is required, delete the VCI using DROP INDEX, and re-create it using CREATE INDEX after completing the operation.
	SET TABLESPACE	
	ALL IN TABLESPACE	
ALTER OPERATOR CLASS	RENAME TO	The subcommands on the left cannot be performed if a VCI is specified. This operation is not supported in VCI.
	OWNER TO	
	SET SCHEMA	
ALTER OPERATOR FAMILY	ADD	
	DROP	
	RENAME TO	
	OWNER TO	
	SET SCHEMA	
ALTER TABLE	ALL IN TABLESPACE *name* [OWNED BY *roleName*] SET TABLESPACE *newTablespace*	A tablespace that contains a VCI cannot be specified. If the operation is required, delete the VCI using DROP INDEX, and re-create it using CREATE INDEX after completing the operation.
	DROP [COLUMN] [IF EXISTS] *colName* [RESTRICT | CASCADE]	A column that contains a VCI cannot be specified. If the operation is required, delete the VCI using DROP INDEX, and re-create it using CREATE INDEX after completing the operation.
	ALTER [COLUMN] *colName* [SET DATA] TYPE *dataType* [COLLATE *collation*] [USING *expr*]	

Command	Subcommand	Description
	CLUSTER ON *indexName*	A VCI cannot be specified. This operation is not supported in VCI.
	REPLICA IDENTITY {DEFAULT | USING INDEX *indexName* | FULL | NOTHING}	
CLUSTER	-	A table that contains a VCI and VCI cannot be specified. If the operation is required, delete the VCI using DROP INDEX, and re-create it using CREATE INDEX after completing the operation.
CREATE INDEX	UNIQUE	The subcommands on the left cannot be performed if a VCI is specified. This operation is not supported in VCI.
	CONCURRENTLY	
	[ASC | DESC]	
	[NULLS { FIRST | LAST }]	
	WITH	
	WHERE	
	INCLUDE	
CREATE OPERATOR CLASS	-	A VCI cannot be specified. This operation is not supported in VCI.
CREATE OPERATOR FAMILY	-	
CREATE TABLE	EXCLUDE	
DROP INDEX	CONCURRENTLY	The subcommands on the left cannot be performed if a VCI is specified. This operation is not supported in VCI.
REINDEX	-	A VCI cannot be specified. This command is not required as VCI uses daemon's automatic maintenance to prevent disk space from increasing.

Client application command

- Operations that cannot be performed on relations containing a VCI

Command	Description
clusterdb	Clustering cannot be performed for tables that contain a VCI.
reindexdb	VCIs cannot be specified on the --index option.

9.2.2 Data Preload Feature

The first aggregation using VCI immediately after an instance is started may take time, because the VCI data has not been loaded to buffer.

Therefore, use the preload feature to load the VCI data to buffer in advance when performing VCI aggregation after an instance is started.

When using the preload feature, execute the function pgx_prewarm_vci to each VCI created with CREATE INDEX.

```
db01=# SELECT pgx_prewarm_vci('idx_vci');
```

See

Refer to "B.4 VCI Data Load Control Function" for information on pgx_prewarm_vci.

Chapter 10 Parallel Query

Fujitsu Enterprise Postgres enhances parallel queries, by taking into consideration the aspects below:

- CPU load calculation
- Increase of workers during runtime
- Statistics view displays the action state

10.1 CPU Load Calculation

There may be a case when the user tries to execute a parallel query but there is not enough CPU available.

Adding dynamic workers at this stage will provide no benefits - instead, it may add overhead due to context switching.

Fujitsu Enterprise Postgres takes into consideration the current CPU load when deciding on the number of workers for parallel query.

10.2 Increase of Workers during Runtime

This Fujitsu Enterprise Postgres enhancement allows systems to allocate additional workers during query execution (if there are workers available at the time). This improves query performance, which could otherwise starve of CPU if there were fewer or no workers when the query started.

Note

The ability to increase workers during runtime is available only with parallel query.

10.3 Statistics View Displays the Action State

You can check the action state by using the existing statistics collector. Monitoring the statistics enables you to check the action state of parallel processing through, for example, changes in the system load.

The following table describes the column added to the statistics views:

pg_stat_all_tables

Column	Data type	Description
parallel_query	bigint	Number of parallel query initialized in the table.

The parallel query column has also been added to the following views:

- pg_stat_sys_tables
- pg_stat_user_tables
- pg_stat_xact_all_tables
- pg_stat_xact_sys_tables
- pg_stat_xact_user_tables

See

Refer to "The Statistics Collector" in "Server Administration" in the PostgreSQL Documentation for information on the statistics collector and views.

Values will be set for the pgx_stat_sql view in Fujitsu Enterprise Postgres if parallel query is run.

See

Refer to "C.5 pgx_stat_sql" for information on pgx_stat_sql view.

Chapter 11 High-Speed Data Load

High-speed data load uses the pgx_loader command to load data from files at high speed into Fujitsu Enterprise Postgres.

Note

- This feature is available only in the Advanced Edition.
- This feature is not available in single-user mode. This is because in single-user mode instances run in a single process, and it cannot start parallel workers.

11.1 Installing High-Speed Data Load

This section describes how to install high-speed data load.

Installation flow

1. Deciding whether to Install
2. Estimating Resources
3. Setup

11.1.1 Deciding whether to Install

The feature achieves high speed data load by executing the COPY FROM command in parallel. If the database system is unable to use sufficient resources due to the feature using more resources than the COPY FROM command of PostgreSQL, load performance may be inferior to that of the COPY FROM command of PostgreSQL. Therefore, determine if the feature will be effective by considering the factors below before deciding to install.

Database server memory

If the value of shared_buffers in postgresql.conf is small, fewer data pages are cached to the shared memory of the database server. This will result in multiple parallel workers more often having to wait for write exclusive locks to the same data page. Moreover, the smaller the number of data pages, the more often the table expands. During table expansion, access to the table is exclusive (standby event name: extend), so write time increases. To cater for that, increase the value of shared_buffers.

See

The standby event name is stored in the wait_event column of the pg_stat_activity view. Refer to "wait_event Description" in "The Statistics Collector" in the PostgreSQL Documentation for details.

Frequency of checkpoints

If checkpoints are issued at short intervals, write performance is reduced. If the messages below are output to the server log during data writes, increase the values of max_wal_size and checkpoint_timeout in postgresql.conf to reduce the frequency of checkpoints.

Example

```
LOG:  checkpoints are occurring too frequently (19 seconds apart)
HINT:  Consider increasing the configuration parameter "max_wal_size".
```

11.1.2 Estimating Resources

Estimate the memory requirements for high-speed data load.

Up to 128 parallel workers to perform data load can be specified for this feature. The additional resources below are required depending on the number of parallel workers.

- Dynamic shared memory created during data load

  The feature creates shared memory and shared memory message queues during data load. These are used to send external data from the back end to the parallel workers, and for error notifications.

Note

If the value of shared_buffers in postgresql.conf is small, the system will often have to wait for write exclusive locks to the same data page (as described in "Database server memory" in "11.1.1 Deciding whether to Install"). Since input data cannot be loaded from the shared memory message queues during such waits, they will often be full. In these cases, it will not be possible to write to the shared memory message queues, resulting in degraded data load performance.

See

Refer to "High-Speed Data Load Memory Requirements" in the Installation and Setup Guide for Server for information on the formula for estimating memory requirements.

11.1.3 Setup

This section describes how to set up high-speed data load.

Setup flow

1. Setting Parameters
2. Installing the Extension

11.1.3.1 Setting Parameters

Set the parameters required for high-speed data load in postgresql.conf. After that, start or restart the instance.

The table below lists the postgresql.conf parameters that must be changed, and the values that must be added to their current values. After editing postgresql.conf, start or restart the instance.

Parameter	Setting	Required
max_prepared_transactions	Add the number of transactions that can be prepared by parallel workers during data load to the parameter's current value. The resulting value must be equal to or greater than the maximum number of parallel workers used with this feature.	Mandatory
max_worker_processes	Number of parallel workers to perform data load.	Mandatory
max_parallel_workers	Add the maximum number of parallel workers to be used in a parallel query by this feature to the parameter's current value. The resulting value must be equal to or greater than the number of parallel workers used with this feature.	Mandatory

Example

The example below shows how to configure 2 instances of high-speed data load being executed simultaneously using a degree of parallelism of 4.

```
max_prepared_transactions = 13  #Example if the initial value was 5: 5 + 2 x 4 = 13
max_worker_processes = 16        #Example if the initial value was 8: 8 + 2 x 4 = 16
max_parallel_workers = 12        #Example if the initial value was 4: 4 + 2 x 4 = 12
```

Note

As shown in the example above, set the value of max_prepared_transactions, max_worker_processes and max_parallel_workers multiplied by the number of instances of this feature executed simultaneously.

The table below lists the postgresql.conf parameters that must also be checked.

Parameter	Setting	Required
dynamic_shared_memory_type	Implementation of dynamic shared memory to be used by the instance. The default value is recommended.	Mandatory

See

Refer to "Resource Consumption" in the PostgreSQL Documentation for information on the parameters.

11.1.3.2 Installing the Extension

Execute CREATE EXTENSION to install the high-speed data load extension. The extension needs to be installed on each database.

Example

The example below installs the extension on the "postgres" database.

```
postgres=# CREATE EXTENSION pgx_loader;
CREATE EXTENSION
```

Note

- Only superusers can install the high-speed data load extension.
- The high-speed data load extension can only be installed on the public schema.

11.2 Using High-Speed Data Load

This section describes how to use high-speed data load.

11.2.1 Loading Data

To load data from a file into a Fujitsu Enterprise Postgres table, execute the pgx_loader command in load mode.

Example

The example below loads the file E:\path\to\data.csv (2000 records) into table tbl using a degree of parallelism of 3.

```
> pgx_loader load -j 3 -c "COPY tbl FROM 'E:\path\to\data.csv' WITH CSV"
LOAD 2000
```

Point

If an external file contains data that violates the format or constraints, the data load may fail partway through, resulting in delays for routine tasks such as nightly batch processing. Therefore, it is recommended to remove the invalid data before executing the data load.

Note

The data inserted using this feature is dumped as a COPY command by the pg_dump command and the pg_dumpall command.

See

- Refer to "pgx_loader" in the Reference for information on the command.
- Refer to "COPY" in the PostgreSQL Documentation for information on the deployment destination and access privileges for external files.

11.2.2 Checking Progress

If you are performing a data load with a large external file as input, you can verify that the process is continuing by getting progress information during the load. Progress information can be obtained from the pgx_stat_progress_loader view. This view displays the sum of the progress information of the back-end process and the number of parallel worker processes. Search the pgx_stat_progress_loader view, for example, with a SELECT statement, to locate the appropriate row. After running the pgx_loader command, look in the pg_stat_activity view and locate a row in the pgx_stat_progress_loader view with the PID obtained.

Example

1. See the pg_stat_activity view. (9311 for back-end processes, 9312, 9313, 9314 for worker processes)

```
postgres=# select pid, application_name, backend_type from pg_stat_activity
pid  | application_name |         backend_type
-----+------------------+------------------------------
6216 |                  | autovacuum launcher
6218 |                  | logical replication launcher
6271 | psql             | client backend
9311 | pgx_loader       | client backend
9312 |                  | parallel loader for PID 9311
9313 |                  | parallel loader for PID 9311
9314 |                  | parallel loader for PID 9311
6214 |                  | background writer
6213 |                  | checkpointer
6215 |                  | walwriter
```

2. Check the information in the pgx_stat_progress_loader view.

```
postgres=# SELECT * FROM pgx_stat_progress_loader
pid  | datid | datname | relid | command  | type | bytes_processed | bytes_total |
tuples_processed | tuples_excluded
-----+-------+---------+-------+----------+------+-----------------+-------------
+------------------+----------------
9311 |  222  | testdb  | 333   |COPY FROM | FILE |        192000   |    450000   |        189000
|           3000
```

Refer to "C.7 pgx_stat_progress_loader" for information on the pgx_stat_progress_loader view.

Note

When you run the pgx_loader command, the PostgreSQL pg_stat_progress_copy view prints the progress of the back-end process and the number of parallel worker processes on each line. The backend process progress information tuples_processed, tuples_excluded is 0. Also, bytes_processed and bytes_total for worker processes are 0.

```
postgres=# SELECT * FROM pg_stat_progress_copy
pid  | datid | datname | relid | command  | type | bytes_processed | bytes_total | tuples_processed |
tuples_excluded
-----+-------+---------+-------+----------+------+-----------------+-------------+------------------
+---------------
9311 |  222  | testdb  | 333   |COPY FROM | FILE |         192000  |    450000   |              0
|               0
9312 |  222  | testdb  | 333   |COPY FROM | FILE |              0  |         0   |          63000
|            1000
9313 |  222  | testdb  | 333   |COPY FROM | FILE |              0  |         0   |          63000
|            1000
9314 |  222  | testdb  | 333   |COPY FROM | FILE |              0  |         0   |          63000
|            1000
```

Refer to "pg_stat_progress_copy View" in the PostgreSQL Documentation for information on the pg_stat_progress_copy view.

11.2.3 Recovering from a Data Load that Ended Abnormally

If a system interruption such as a server failure occurs while high-speed data load is being performed, transactions prepared using this feature may be changed to the in-doubt state. At that point, resources occupied by the transaction will be locked, and access to the relevant resources from other transactions will be blocked, rendering them unusable.

In such cases, check transactions that are in an in-doubt state, and resolve them.

Checking for in-doubt transactions

This section describes how to check for in-doubt transactions.

1. Refer to the pgx_loader_state table in the pgx_loader schema.

   Retrieve the global transaction identifier (gid column) of in-doubt transactions. In-doubt transactions will contain "rollback" in the column "state".

   #### Example

   The example below retrieves the global transaction identifier (gid) of in-doubt transactions performed by the database role myrole and that used table tbl. The retrieved global transaction identifiers pgx_loader:9589 and pgx_loader:9590 identify in-doubt transactions.

   ```
   postgres=# SELECT gid, state FROM pgx_loader.pgx_loader_state WHERE
   postgres-# role_oid IN (SELECT oid FROM pg_roles WHERE rolname = 'myrole') AND
   postgres-# relation_oid IN (SELECT relid FROM pg_stat_all_tables WHERE
   postgres-# relname = 'tbl');
          gid       | state
   -----------------+-------------
    pgx_loader:9590 | rollback
    pgx_loader:9591 | commit
    pgx_loader:9589 | rollback
   (3 rows)
   ```

2. Refer to the pg_prepared_xacts system view.

   Check if the in-doubt transactions retrieved above exist.

Example

The example below checks if in-doubt transactions with the global transaction identifiers pgx_loader:9589 and pgx_loader:9590 exist.

```
postgres=# SELECT gid FROM pg_prepared_xacts WHERE gid IN ('pgx_loader:9589','pgx_loader:9590');
       gid
------------------
 pgx_loader:9590
 pgx_loader:9589
(2 rows)
```

See

Refer to "E.1 pgx_loader_state" for information on the pgx_loader_state table.

Resolving in-doubt transactions

Execute the pgx_loader command in recovery mode to resolve in-doubt transactions.

After executing the pgx_loader command in recovery mode, perform the procedure described in "Checking for in-doubt transactions" to check if the in-doubt transactions have been resolved.

Example

The example below completes the in-doubt transactions prepared for table tbl.

```
$ pgx_loader recovery -t tbl
```

Point

The recovery mode of the pgx_loader command only resolves transactions prepared by high-speed data load. For transactions prepared by an application using distributed transactions other than this feature, follow the procedure described in "15.13 Actions in Response to Error in a Distributed Transaction".

11.3 Removing High-Speed Data Load

This section describes how to remove high-speed data load.

11.3.1 Removing the Extension

Execute DROP EXTENSION to remove the high-speed data load extension. The extension needs to be removed on each database.

Example

The example below removes the extension on the "postgres" database.

```
postgres=# DROP EXTENSION pgx_loader;
DROP EXTENSION
```

Note

- The information required for operation of high-speed data load is stored in the pgx_loader_state table of the pgx_loader schema. Do not remove the high-speed data load extension if the pgx_loader_state table is not empty.
- Only superusers can remove the high-speed data load extension.
- The high-speed data load extension can only be removed on the public schema.

Chapter 12 Global Meta Cache

The Global Meta Cache (GMC) feature loads a meta cache into shared memory using the pgx_global_metacache parameter. This reduces the amount of memory required throughout the system.

Note

This feature can only be used in Advanced Edition.

12.1 Usage

Describes how to use the Global Meta Cache feature.

12.1.1 Deciding Whether to Enable the Global Meta Cache Feature

Global Meta Cache is a mechanism for sharing meta caches between processes, so it works well on systems with a high number of resources accessed and SQL connections. The number of resources is primarily the number of tables accessed by a process, the number of indexes, or the total number of all columns in all tables accessed.

In particular, consider using Global Meta Cache if the total size of the meta cache for each process exceeds the amount of installed memory, or takes up a large portion of that memory, thereby squeezing memory allocations to the database cache or the Operating system file cache. Using Global Meta Cache may increase the time it takes to execute a single SQL to reference a meta cache on shared memory, but you can expect a greater benefit from being able to allocate more memory, such as for the database cache.

If performance degradation using Global Meta Cache is not acceptable, you may want to limit the number of tables accessed by a process.

12.1.2 Estimating Memory for Global Meta Cache

To enable the Global Meta Cache feature, the pgx_global_metacache parameter must specify an upper limit on the size of the shared memory (Hereinafter, the GMC area) dedicated to Global Meta Cahche. Ideally, this upper limit should be the size estimated in “Appendix A Parameters”. Values lower than this can still work, but refer to "12.1.3 How the GMC Memory Area Is Used" on using the GMC area to understand the disadvantages.

12.1.3 How the GMC Memory Area Is Used

At startup, the memory for the GMC area is not used much, but the GMC area grows as new meta caches are placed in the GMC area. If it does, it discards any meta caches that the system determines are not heavily used and places a new one in the GMC area.

Therefore, the GMC area will work even if it is smaller than the estimate, but the meta cache will be regenerated if the discarded meta cache needs to be reused. Note that if this happens frequently, it will degrade overall performance.

With this in mind, it may not be a problem if, for example, the tables to be accessed are different depending on the time zone, and the degradation of the time zone immediately after the change is acceptable.

In any case, be sure to test and tune the system thoroughly before running it.

12.1.4 Enabling the Global Meta Cache Feature

To Enable the Global Meta Cache feature edit the postgresql.conf file and set the pgx_global_metacahe parameter. Restarting the instance after editing the postgresql.conf file is required. Refer to "Appendix A Parameters" for information on the parameters.

Parameter Name	Description
pgx_global_metacache	Specify the maximum amount of memory for the GMC area on shared memory. When it’s set to 0 (default value), the Global Meta Cache feature is disabled. When enabled, the minimum value allowed is 10MB.

When the cache is created, if the total amount of meta caches on shared memory exceeds the value specified by pgx_global_metacache, the inactive, unreferenced meta caches are removed from the GMC area. Note that if all GMC are in use and the cache cannot be created in the GMC area, the cache is temporarily created in the local memory of the backend process.

Example

Here is an example postgresql.conf configuration:

```
pgx_global_metacache = 800 MB
```

Wait Events

The Global Meta Cache feature may cause wait events. Wait events are identified in the wait_event column of the pg_stat_activity view. GMC specific wait events are described below.

[GMC Feature Wait Events]

Wait Event Type	Wait Event Name	Description
LWLock	GlobalCatcache	Waiting to find, add, and remove meta caches in the GMC area.
IPC	GMCSweep	Waiting to select a meta cache that can be deleted when GMC space is low. If the GMC is fully referencing and there is no deletable meta cache, it is waiting for the reference to be removed and a deletable meta cache to be selected.

Note

If GMCSweep is happened frequently, increase the pgx_global_metacache setting.

See

Refer to "Viewing Statistics" in the PostgreSQL Documentation for information on the pg_stat_activity view.

12.1.5 Estimating Resources

Refer to "Global Meta Cache Memory Requirements" in the Installation and Setup Guide for Server for formulas to estimate the amount of memory used by the Global Meta Cache feature.

12.2 Statistics

Describes the statistics for the Global Meta Cache feature.

12.2.1 System View

You can check the cache hit ratio and size of the GMC area in the system view pgx_stat_gmc. Refer to "C.6 pgx_stat_gmc" for information on the columns.

If the cache hit ratio is low and the current memory usage is close to pgx_global_metacache, increase the pgx_global_metacache setting because performance may be degraded.

Refer to "7.6 Monitoring Database Activity" in the Operations Guide for information on the statistics.

Chapter 13 Local Meta Cache Limit

Local Meta Cache Limit feature limits the size of a Local Meta Cache by removing it if it has not been accessed for a long time.

Note

This feature is available only in the Advanced Edition.

13.1 Usage

Describes how to use the Local Meta Cache Limit feature.

13.1.1 Deciding Whether to Enable the Local Meta Cache Limit Feature

Refer to "Appendix A Parameters", after estimating the total amount of memory to be used as the catalog cache and relation cache, when the total amount of memory exceeds the amount of installed memory or occupies a large amount of installed memory, consider using this feature.

This feature adds the action of discarding the meta cache that has been held permanently. If you attempt to refer to a destroyed meta cache again, the meta cache is recreated, so using this feature will result in poor performance compared to not using it.

Therefore, read the following to understand how to discard a meta cache.

- 13.1.3 Cache Removal when Local Meta Cache Limit is Enabled
- 13.1.4 Performance Impact and Parameter Tuning of the Local Meta Cache Limit Feature
- Parameters for the Local Meta Cache Limit feature

How to set the upper limit with these considerations is described in detail in the estimation formula in "Appendix A Parameters".

13.1.2 How to Set Parameters for the Local Meta Cache Limit Feature

To enable the Local Meta Cache Limit feature, set the pgx_catalog_cache_max_size and pgx_relation_cache_max_size parameters.

Parameter Name	Description
pgx_catalog_cache_max_size	Specify the maximum amount of memory that the backend process should use as the catalog cache. You can enable catalog cache removal by setting it to 8 KB or more. When it is set to 0 (default value), the catalog cache removal is disabled.
pgx_relation_cache_max_size	Specify the maximum amount of memory that the backend process should use as the relation cache. You can enable relation cache removal by setting it to 8 KB or more. When it is set to 0 (default value), the relation cache removal is disabled.

Example

Here is an example postgresql.conf configuration:

```
pgx_catalog_cache_max_size = 1MB
pgx_relation_cache_max_size = 1MB
```

13.1.3 Cache Removal when Local Meta Cache Limit is Enabled

When this feature is enabled, the caching strategy is to keep the cache as long as possible within the specified upper limit. If holding a new cache exceeds the limit, consider locality of reference and remove the cache from the one with the longest unreferenced time.

However, because the cache used by active transactions cannot be removed, if a transaction uses a large number of caches, the cache may be held above the limit. In this case, remove the all caches at the end of the transaction. This is necessary to free up memory.

In PostgreSQL, in order to acquire memory at high speed, a memory block of a certain size is acquired from the OS, and a small memory is cut out from the block and used. The memory for the metacache is cut out in the same way. Therefore, it is possible to return the memory block to the OS by destroying all the meta caches scattered throughout the memory block. When this happens, the next SQL execution will be slowed down due to the re-creation of the meta cache. Therefore, upper limit of feature should be set to a value larger than the size of the meta cache used by at least one transaction.

When the size of the meta cache exceeds the upper limit, the following message is output:

```
WARNING: could not reduce Cat/RelCacheMemoryContext size to AA kilobytes, reduced to BB kilobytes
HINT: consider increasing the configuration parameter pgx_catalog/relation_cache_max_size
```

(***AA***: Upper limit, ***BB***: Amount of memory actually used)

CatCacheMemoryContext and RelCacheMemoryContext are memory areas for storing the catalog cache and relation cache, respectively. If this message is output, consider increasing the upper limit.

If the memory consumption by the backend process exceeds the allowable value by increasing the upper limit, reconsider the SQL to be executed, such as reducing the number of tables accessed in one transaction, or add memory adjust to the amount of memory used.

13.1.4 Performance Impact and Parameter Tuning of the Local Meta Cache Limit Feature

By observing how much meta cache regeneration is taking place, you can determine if the low upper limit is the cause of the failure to achieve the desired performance.

From the message below, calculate the cache hit ratio as follows:

```
Cache hit ratio = Number of cache hits ÷ Number of times the cache was searched
```

If the cache hit ratio is 80% or higher, this feature will not be the main factor that impedes performance. If not, raise the upper limit and see if performance can reach the goal. In doing so, first try to shift the focus of allocation to the relations cache. This is because when executing SQL, the relation cache generated based on the catalog cache is mainly referenced, so it is advantageous to leave a large amount of relation cache.

```
Catalog cache:catalog cache hit stats: search XX, hits YY
Relation cache:relation cache hit stats: search XX, hits YY
```

(***XX***: Number of times the cache was searched, ***YY***: Number of cache hits)

This message is printed when the transaction ends. However, if you output the message frequently, the performance will be degraded by itself, so you can adjust the output interval with the following parameters.

Parameter Name	Description
pgx_cache_hit_log_interval	When the transaction ends, if the time set in this parameter has elapsed since the previous message was output, the message is output. If set to 0, a message will be output each time the transaction ends. Setting -1 disables the output. The default value is 10min. Even if pgx_catalog_cache_max_size and pgx_relation_cache_max_size are disabled, the message output of the corresponding cache will be invalid. Immediately after connecting to the server, a small transaction occurs before the request from the user application, such as for user authentication. Since it is meaningless to know the hit ratio for these, a message is output at the end of the transaction that started after the time set in this parameter has elapsed after connecting to the server. For the same reason, setting a small value such as 0 may result in a message being printed at the end of such a small transaction. You can check which transaction the message corresponds to from the information output at the beginning. This information depends on the setting of the parameter log_line_prefix.

Example

Here is an example postgresql.conf configuration:

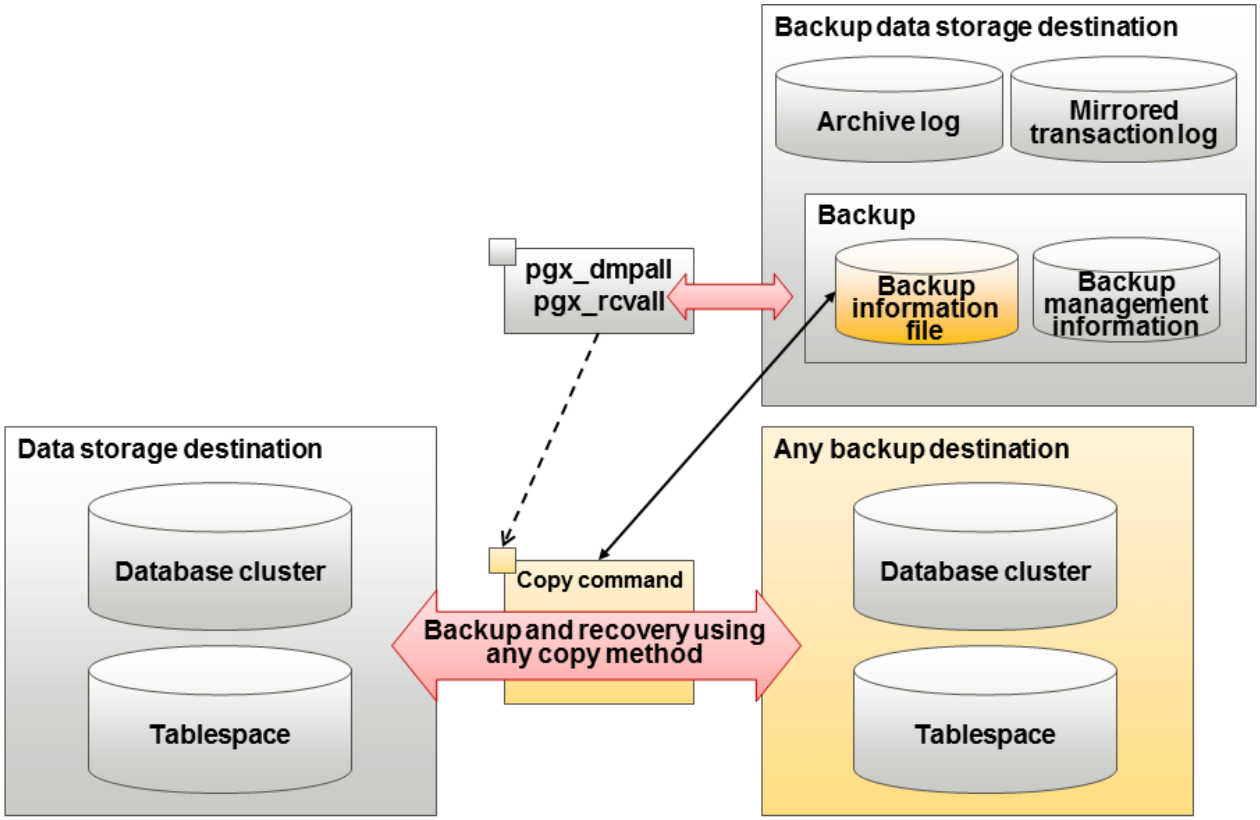
```
pgx_cache_hit_log_interval= 30min
```

Chapter 14 Backup/Recovery Using the Copy Command

By using a copy command created by the user, the pgx_dmpall command and the pgx_rcvall command can perform backup to any destination and can perform recovery from any destination using any copy method.

Copy commands must be created in advance as executable scripts for the user to implement the copy process on database clusters and tablespaces, and are called when executing the pgx_dmpall and pgx_rcvall commands.

This appendix describes backup/recovery using the copy command.



Point

It is also possible to back up only some tablespaces using the copy command. However, database resources not backed up using the copy command are still backed up to the backup data storage destination.

Note

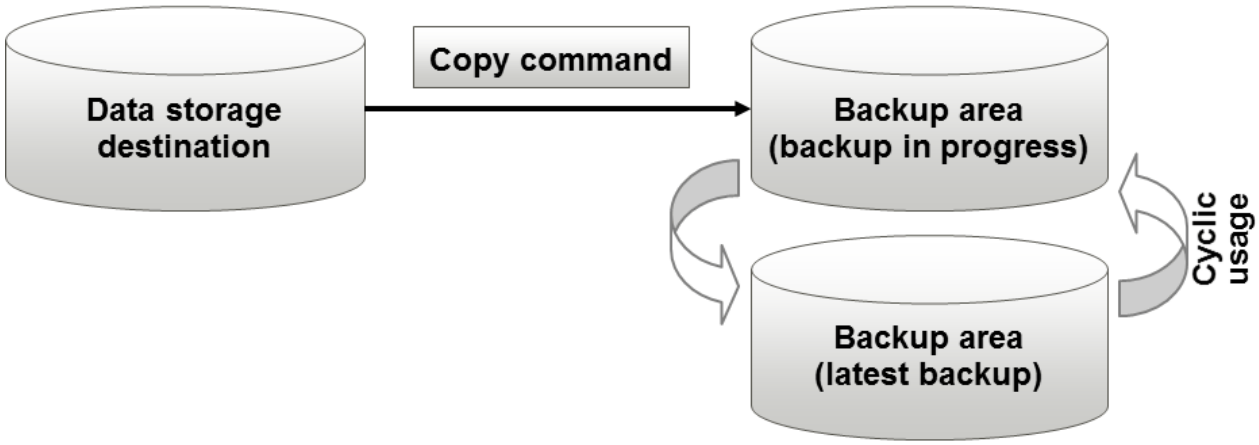
Both the backup data storage destination and the optional backup destination are necessary for recovery - if they are located in secondary media, combined management of these is necessary.

14.1 Configuration of the Copy Command

This section describes the configuration of the copy command for backup and recovery.

Cyclic usage of the backup area

Prepare two backup areas for the copy command in case an issue affects the data storage destination during backup. The copy command performs backup while cyclically using these backup areas.

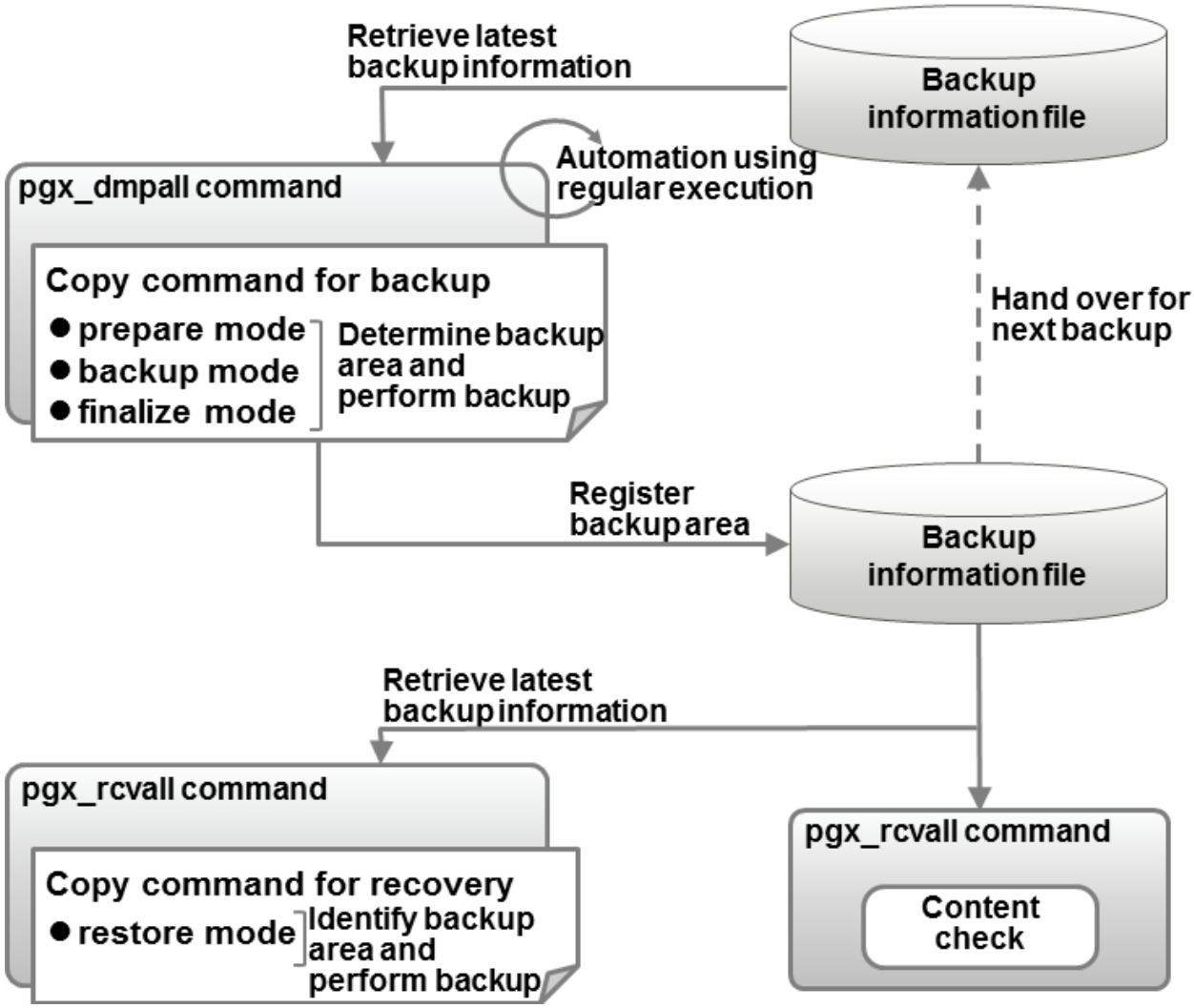


Note

The backup data storage destination cannot be used as these backup areas used by the copy command.

Backup using the backup information file

The copy command must determine the backup destination on each backup, as it is necessary to cycle through the backup areas. Backup can be automated by using the backup information file, which contains information about the backup destination.



Information

The backup information file is prepared in the backup data storage destination by the pgx_dmpall command, and contains information that can be read or updated by the copy command. This file is managed by associating it with the latest backup successfully completed by the pgx_dmpall command, so the latest backup information relating to the copy command registered by the user can be retrieved. Additionally, the content of the backup information file can be displayed using the pgx_rcvall command.

Configuration of the copy command for backup

The pgx_dmpall command calls the copy command for backup after execution for the three modes below. It is therefore necessary for the copy command for backup to implement the required processing for each of the modes.

- prepare mode

  Determines which of the two backup areas will be used for the current backup.

  The backup area to be used for the current backup is determined by reading the information relating to the latest backup destination where the backup information file was written to during the previous backup.

- backup mode

  Performs backup on the backup area determined by prepare mode, using any copy method.

- finalize mode

  Writes information relating to the destination of the current backup to the backup information file.

  This enables the prepare mode to check the destination of the previous backup during the next backup.

Note

The user can use any method to hand over backup information between modes within the copy command, such as creating temporary files.

Configuration of the copy command for recovery

The pgx_rcvall command calls the copy command for recovery for the mode below. It is therefore necessary for the copy command for recovery to implement the required processing for the mode.

- restore mode

  Any copy method can be used to implement restore from the backup destination retrieved using the copy command for backup.

Point

By referring to the mode assigned to the copy command as an argument, backup and recovery can be implemented using a single copy command.

Example

Using a batch file

```
if "%1" == "prepare"  goto prepare
if "%1" == "backup"   goto backup
if "%1" == "finalize" goto finalize
if "%1" == "restore"  goto restore
:prepare
    processingRequiredForPrepareMode
    exit /b
:backup
    processingRequiredForBackupMode
```

```
    exit /b
:finalize
    processingRequiredForFinalizeMode
    exit /b
:restore
    processingRequiredForRestoreMode
    exit /b
```

Point

A sample batch file that backs up the database cluster and tablespace directory to a specific directory is supplied to demonstrate how to write a copy command.

The sample is stored in the directory below:

```
installDir\share\copy_command.bat.sample
```

14.2 Backup Using the Copy Command

To perform backup using the copy command, in addition to performing the standard backup procedure, it is also necessary to create a copy command, and then execute the pgx_dmpall command specifying it. This section describes the procedure specific to using the copy command.

Preparing for backup

You must prepare for backup before actually starting the backup process.

Perform the following procedure:

1. Determine the database resources to be backed up

   Determine the database resources to be backed up using the copy command. The copy command can back up the following resources:

   - Database cluster
   - Tablespace

   To back up only some tablespaces, create a file listing them. This file is not necessary to back up all tablespaces.

   Example

   To back up only tablespaces tblspc1 and tblspc2

   ```
   tblspc1
   tblspc2
   ```

2. Prepare a backup area

   Prepare a backup area to save the database resources to be backed up, as determined in step 1.

3. Create the copy command

   Create the copy commands for backup and recovery. Refer to "14.4 Copy Command Interface" for details.

Performing backup

Execute the pgx_dmpall command with the -Y option specifying the full path of the copy command for backup created in step 3 of preparation for backup.

The example below backs up only some tablespaces, but not the database cluster, using the copy command.

Example

```
> pgx_dmpall -D D:\database\inst1 -Y D:\database\command\backup.bat
  --exclude-copy-cluster -P D:\database\command\tablespace_list.txt
```

Point

- To exclude up the database cluster from backup using the copy command, specify the --exclude-copy-cluster option.
- To back up only some tablespaces using the copy command, use the -P option specifying the full path of the file created in step 1 of preparation for backup.

See

- Refer to "pgx_dmpall" in the Reference for information on the command.

Checking backup status

Use the pgx_rcvall command to check the backup status.

Execute the pgx_rcvall command with the -l option specified to output backup data information. If backup was performed using the copy command, the resources backed up using the copy command will also be output.

Example

```
> pgx_rcvall -l -D D:\database\inst1
Date                Status   Dir                                 Resources backed up by the copy command
2022-03-01 13:30:40 COMPLETE E:/backup/inst1/2022-03-01_13-30-40 pg_data,dbspace,indexspace
```

14.3 Recovery Using the Copy Command

To perform recovery using the copy command, in addition to performing the standard recovery procedure, it is also necessary to create a copy command, and then execute the pgx_rcvall command specifying it. This section describes the procedure specific to using the copy command.

Determining the backup area of the latest backup

Check the backup information file to determine the backup area used for the latest backup, and confirm that it is in a recoverable state.

Execute the pgx_rcvall command with the --view-results-of-copying option to output the content of the backup information file.

Example

```
> pgx_rcvall -D D:\database\inst1 --view-results-of-copying
```

Perform recovery

Execute the pgx_rcvall command with the -Y option specifying the full path of the copy command for recovery created in step 3 of the preparation for backup described in "14.2 Backup Using the Copy Command".

The example below recover only some tablespaces, but not the database cluster, using the copy command.

Example

```
> pgx_rcvall -D D:\database\inst1 -B E:\backup\inst1 -Y D:\database\command\recovery.bat
```

Point

If the latest backup was performed using the copy command, the pgx_rcvall command automatically recognizes which database resources were backed up using the copy command, or whether resources were backed up to the backup data storage destination. Therefore, recovery can be performed by simply executing the pgx_rcvall command specifying the copy command for recovery.

See

Refer to "pgx_rcvall" in the Reference for information on the command.

14.4 Copy Command Interface

The following types of copy command are available:

- Copy command for backup
- Copy command for recovery

This appendix describes the interface of each copy command.

14.4.1 Copy Command for Backup

Feature

User exit (for the copy command) called from the pgx_dmpall command.

Format

The syntax for calling the copy command from the pgx_dmpall command is described below.

If the operation mode is "prepare"

```
copyCommandName prepare "pathOfBackupInfoFile" "pathOfBackupTargetListFile"
```

If the operation mode is "backup"

```
copyCommandName backup
```

If the operation mode is "finalize"

```
copyCommandName finalize "pathOfBackupInfoFile"
```

Argument

- Operation mode

Mode	Description
prepare	Implements the preparation process for backing up using the copy command. Called before the PostgreSQL online backup mode is started.
backup	Implements the backup process. Called during the PostgreSQL online backup mode.

Mode	Description
finalize	Implements the backup completion process. Called after the PostgreSQL online backup mode is completed.

- Full path of the backup information file

  Full path of the backup information file of the latest backup, enclosed in double quotation marks. If a backup has not been performed, specify '-'.

- Full path of the backup target list file

  Full path of the file containing the resources to be backed up using the copy command, enclosed in double quotation marks. One of the following is described in each resource name.

Resource	Description
Database cluster	pg_data
Tablespace	Tablespace name

Example

To back up the database cluster and the tablespaces dbspace and indexspace using the copy command, the file should contain the following:

```
pg_data
dbspace
indexspace
```

Information

The encoding of resource names output to the backup target list file by the pgx_dmpall command is the encoding used when this command connects to the database with auto specified for the client_encoding parameter, and is dependent on the locale at the time of command execution.

The number of arguments vary depending on operation mode. The argument of each operation mode is as follows.

Operation mode	First argument	Second argument	Third argument
prepare	Operation mode	Backup information file path name	Backup target list file path name
backup		None	None
finalize		Backup information file path name	

Additionally, the access permissions for the backup information file and backup target list file are different depending on the operation mode. The access permissions of each operation mode are as follows:

Operation mode	Backup information file	Backup target list file
prepare	Can be viewed	Can be viewed
backup	-	-
finalize	Can be viewed and updated	-

Return value

Return value	Description
0	Normal end

Return value	Description
	The pgx_dmpall command continues processing.
Other than 0	Abnormal end The pgx_dmpall command terminates in error.

Description

- The copy command operates with the privileges of the operating system user who executed the pgx_dmpall command. Therefore, grant copy command execution privileges to users who will execute the pgx_dmpall command. Additionally, have the copy command change users as necessary.
- To write to the backup information file, use a method such as redirection from the copy command.
- Because the copy command is called for each mode, implement all processing for each one.
- To copy multiple resources simultaneously, have the copy command copy them in parallel.

- The backup information file and backup target list file cannot be deleted. Additionally, the privileges cannot be changed.
- Standard output and standard error output of the copy command are output to the terminal where the pgx_dmpall command was executed.
- If the copy command becomes unresponsive, the pgx_dmpall command will also become unresponsive. If the copy command is deemed to be unresponsive by the operating system, use an operating system command to forcibly stop it.
- Output the copy command execution trace and the result to a temporary file, so that if it terminates in error, the cause can be investigated at a later time.
- For prepare mode only, it is possible to use the PostgreSQL client application to access the database using the copy command. For all other modes, do not execute Fujitsu Enterprise Postgres commands or PostgreSQL applications.
- Enable the fsync parameter in postgresql.conf, because data on the shared memory buffer needs to have been already written to disk when backup starts.

14.4.2 Copy Command for Recovery

Feature

User exit (for the copy command) called from the pgx_rcvall command.

Format

The syntax for calling the copy command from the pgx_rcvall command is described below.

```
copyCommandName restore "pathOfBackupInfoFile" "pathOfBackupTargetListFile"
```

Argument

- Operation mode

Mode	Description
restore	Performs restore.

- Full path of the backup information file

  Full path of the backup information file, enclosed in double quotation marks.

- Full path of the backup target list file

  Full path of the file containing the resources to be restored using the copy command, enclosed in double quotation marks.

The access permissions for the backup information file and backup target list file are as below.

Backup information file	Backup target list file
Can be viewed	Can be viewed

Return value

Return value	Description
0	Normal end The pgx_rcvall command continues processing.
Other than 0	Abnormal end The pgx_rcvall command terminates in error.

Description

- The copy command operates with the privileges of the operating system user who executed the pgx_rcvall command. Therefore, grant copy command execution privileges to users who will execute the pgx_rcvall command. Additionally, have the copy command change users as necessary.
- The copy command is called once only in restore mode.
- To copy multiple resources simultaneously, have the copy command copy them in parallel.

Note

- The backup information file and backup target list file cannot be deleted. Additionally, the privileges cannot be changed.
- Standard output and standard error output of the copy command are output to the terminal where the pgx_rcvall command was executed.
- If the copy command becomes unresponsive, the pgx_rcvall command will also become unresponsive. If the status of the copy command is deemed to be unresponsive by the operating system, use an operating system command to forcibly stop it.
- Output the copy command execution trace and the result to a temporary file, so that if it terminates in error, the cause can be investigated at a later time.
- Do not execute Fujitsu Enterprise Postgres commands or PostgreSQL applications in the copy command.
- There may be files and directories not required for recovery using the archive log included in the backup, such as postmaster.pid, pg_wal/*subdirectory* and pg_replslot in the database cluster. If such unnecessary files and directories exist, have the copy command delete them after the restore.

Chapter 15 Actions when an Error Occurs

This chapter describes the actions to take when an error occurs in the database or an application, while Fujitsu Enterprise Postgres is operating.

Depending on the type of error, it may be necessary to recover the database cluster. The recovery process recovers the following resources:

- Data storage destination
- Transaction log storage destination (if the transaction log is stored in a separate disk from the data storage destination)
- Backup data storage destination

Note

Even if a disk is not defective, the same input-output error messages, as those generated when the disk is defective, may be output. The recovery actions differ for these error messages.

Check the status of the disk, and select one of the following actions:

- If the disk is defective

  Refer to "15.1 Recovering from Disk Failure (Hardware)", and take actions accordingly.

- If the disk is not defective

  Refer to "15.14 I/O Errors Other than Disk Failure", and take actions accordingly.

  A few examples of errors generated even if the disk is not defective include:

  - Network error with an external disk
  - Errors caused by power failure or mounting issues

Determining the cause of an error

If an error occurs, refer to the WebAdmin message and the event log, and determine the cause of the error.

See

Refer to "Configuring Parameters" in the Installation and Setup Guide for Server for information on server logs.

Approximate recovery time

The formulas for deriving the approximate recovery time of resources in each directory are given below.

If using the copy command with the pgx_rcvall command, the recovery time will depend on the implementation of the copy command.

- Data storage destination or transaction log storage destination

```
Recovery time = (usageByTheDataStorageDestination + usageByTheTransactionLogStorageDestination) /
diskWritePerformance x 1.5
```

  - *usageByTheDataStorageDestination*: Disk space used by the database cluster
  - *usageByTheTransactionLogStorageDestination*: Disk space used by the transaction log stored outside the database cluster
  - *diskWritePerformance*: Measured maximum data volume (bytes/second) that can be written per second in the system environment where the operation is performed
  - 1.5: Coefficient assuming the time excluding disk write, which is the most time-consuming step

- Backup data storage destination

```
Recovery time = usageByTheBackupDataStorageDestination / diskWritePerformance x 1.5
```

- *usageByTheBackupDataStorageDestination*: Disk space used by the backup data
- *diskWritePerformance*: Measured maximum data volume (bytes/second) that can be written per second in the system environment where the operation is performed
- 1.5: Coefficient assuming the time excluding disk write, which is the most time-consuming step

15.1 Recovering from Disk Failure (Hardware)

This section describes how to recover database clusters to a point immediately before failure, if a hardware failure occurs in the data storage disk or the backup data storage disk.

There are two methods of recovery:

- 15.1.1 Using WebAdmin
- 15.1.2 Using Server Command

Point

Back up the database cluster after recovering it. Backup deletes obsolete archive logs (transaction logs copied to the backup data storage destination), freeing up disk space and reducing the recovery time.

15.1.1 Using WebAdmin

Recover the database cluster by following the appropriate recovery procedure below for the disk where the failure occurred.

Note

Recovery operation cannot be performed on an instance that is part of a streaming replication cluster in standby mode.

If disk failure occurs on a standby instance, it may be necessary to delete and re-create the instance.

Recovery operation can be performed on an instance that is part of a streaming replication cluster in "Master" mode. If a recovery operation is performed on a master instance, it will break the replication cluster and streaming replication will stop between the master instance and all its standby instances. In such an event, the standby instances can be promoted to standalone instances or can be deleted and re-created.

If failure occurred in the data storage disk or the transaction log storage disk

Follow the procedure below to recover the data storage disk or the transaction log storage disk.

1. Stop applications

   Stop applications that are using the database.

2. Stop the instance

   Stop the instance. Refer to "2.1.1 Using WebAdmin" for information on how to stop an instance. WebAdmin automatically stops instances if recovery of the database cluster is performed without stopping the instance.

3. Recover the failed disk

   Replace the disk, and then recover the volume configuration information.

4. Create a tablespace directory

   If a tablespace was defined after backup, create a directory for it.

5. Recover the keystore, and enable automatic opening of the keystore

   Do the following if the data in the database has been encrypted:

   - Restore the keystore to its state at the time of the database backup.
   - Enable automatic opening of the keystore.

6. Recover the database cluster

   Log in to WebAdmin, and in the [Instances] tab, click [Solution] for the error message in the lower-right corner.

7. Run recovery

   In the [Restore Instance] dialog box, click [Yes].

   Instance restore is performed. An instance is automatically started when recovery is successful.

8. Resume applications

   Resume applications that are using the database.

Point

WebAdmin may be unable to detect disk errors, depending on how the error occurred.

If this happens, refer to "15.10.3 Other Errors" to perform recovery.

If failure occurred on the backup data storage disk

Follow the procedure below to recover the backup data storage disk.

1. Recover the failed disk

   Replace the disk, and then recover the volume configuration information.

2. Recover the backup data

   Log in to WebAdmin, and in the [Instances] tab, click [Solution] for the error message.

3. Run backup

   Perform backup to enable recovery of the backup data. In the [Backup] dialog box, click [Yes]. The backup is performed. An instance is automatically started when backup is performed.

Point

If you click [Recheck the status], the resources in the data storage destination and the backup data storage destination are reconfirmed. As a result, the following occurs:

- If an error is not detected

  The status of the data storage destination and the backup data storage destination returns to normal, and it is possible to perform operations as usual.

- If an error is detected

  An error message is displayed in the message list again. Click [Solution], and resolve the problem by following the resolution for the cause of the error displayed in the dialog box.

15.1.2 Using Server Command

Recover the database cluster by following the appropriate recovery procedure below for the disk where the failure occurred.

If failure occurred on the data storage disk or the transaction log storage directory

Follow the procedure below to recover the data storage disk or the transaction log storage directory.

1. Stop applications

   Stop applications that are using the database.

2. Stop the instance

   Stop the instance, refer to "2.1.2 Using Commands" for details.

If the instance fails to stop, refer to "15.11 Actions in Response to Failure to Stop an Instance".

3. Recover the failed disk

   Replace the disk, and then recover the volume configuration information.

4. Create a storage destination directory

   - If failure occurred on the data storage disk
     Create a data storage destination directory. If a tablespace was defined, also create a directory for it.
   - If failure occurred on the translation log storage disk
     Create a transaction log storage destination directory.

   In [Properties] in Windows(R) Explorer, set appropriate permissions so that only the instance administrator can access the storage destination directory. (Refer to [Help and Support] in Windows(R) for information on [Properties].)

   See

   Refer to "Preparing Directories to Deploy Resources" under "Setup" in the Installation and Setup Guide for Server for information on how to create a storage directory.

5. Recover the keystore, and enable automatic opening of the keystore

   When the data in the database has been encrypted, restore the keystore to its state at the time of the database backup. Configure automatic opening of the keystore as necessary.

6. Recover the database cluster

   Recover the database cluster using the backup data.

   Specify the following in the pgx_rcvall command:

   - Specify the data storage location in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
   - Specify the backup data storage location in the -B option.

   Example

   ```
   > pgx_rcvall -D D:\database\inst1 -B E:\backup\inst1
   ```

   Note

   If recovery fails, remove the cause of the error in accordance with the displayed error message and then re-execute the pgx_rcvall command.

   If the message "pgx_rcvall: an error occurred during recovery" is displayed, then the log recorded when recovery was executed is output after this message. The cause of the error is output in around the last fifteen lines of the log, so remove the cause of the error in accordance with the message and then re-execute the pgx_rcvall command.

   The following message displayed during recovery is output as part of normal operation of pgx_rcvall command (therefore the user does not need not be concerned).

   ```
   FATAL: the database system is starting up
   ```

7. Start the instance

   Refer to "2.1.2 Using Commands" for information on how to start an instance.

8. Resume applications

   Resume applications that are using the database.

If failure occurred on the backup data storage disk

The procedure for recovering the backup data storage disk is described below.

There are two methods of taking action:

- Performing recovery while the instance is active
- Stopping the instance before performing recovery

The following table shows the different steps to be performed depending on whether you stop the instance.

No	Step	Instance stopped	
		No	Yes
1	Confirm that transaction log mirroring has stopped	Y	N
2	Stop output of archive logs	Y	N
3	Stop applications	N	Y
4	Stop the instance	N	Y
5	Recover the failed disk	Y	Y
6	Create a backup data storage destination directory	Y	Y
7	Resume output of archive logs	Y	N
8	Resume transaction log mirroring	Y	N
9	Start the instance	N	Y
10	Run backup	Y	Y
11	Resume applications	N	Y

Y: Required
N: Not required

The procedure is as follows:

If an instance has not been stopped

1. Confirm that transaction log mirroring has stopped

   Use the following SQL function to confirm that transaction log mirroring has stopped.

```
 postgres=# SELECT pgx_is_wal_multiplexing_paused();
 pgx_is_wal_multiplexing_paused
--------------------
 t
(1 row)
```

   If transaction log mirroring has not stopped, then stop it using the following SQL function.

```
postgres=# SELECT pgx_pause_wal_multiplexing();
LOG:  multiplexing of transaction log files has been stopped
pgx_pause_wal_multiplexing
----------------------------

(1 row)
```

2. Stop output of archive logs

   Transaction logs may accumulate during replacement of backup storage disk, and if the data storage disk or the transaction log storage disk becomes full, there is a risk that operations may not be able to continue.

   To prevent this, use the following methods to stop output of archive logs.

- Changing archive_command

  Specify a command that will surely complete normally, so that archive logs will be regarded as having been output.

  If you specify echo, a message is output to the server log, so it may be used as a reference when you conduct investigations.

- Reload the configuration file

  Execute the pg_ctl reload command or the pg_reload_conf SQL function to reload the configuration file.

If you simply want to stop output of errors without the risk that operations will not be able to continue, specify an empty string ('') in archive_command and reload the configuration file.

3. Recover the failed disk

   Replace the disk, and then recover the volume configuration information.

4. Create a backup data storage destination

   Create a backup data storage destination.

   In [Properties] in Windows(R) Explorer, set appropriate permissions so that only the instance administrator can access the backup data storage destination directory. (Refer to [Help and Support] in Windows(R) for information on [Properties].)

   Refer to "3.2.2 Using Server Commands" for information on how to create a backup data storage destination.

5. Resume output of archive logs

   Return the archive_command setting to its original value, and reload the configuration file.

6. Resume transaction log mirroring

   Execute the pgx_resume_wal_multiplexing SQL function.

   Example

   ```
   SELECT pgx_resume_wal_multiplexing()
   ```

7. Run backup

   Use the pgx_dmpall command to back up the database cluster.

   Specify the following value in the pgx_dmpall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

   Example

   ```
   > pgx_dmpall -D D:\database\inst1
   ```

If an instance has been stopped

1. Stop applications

   Stop applications that are using the database.

2. Stop the instance

   Stop the instance. Refer to "2.1.2 Using Commands" for details.

   If the instance fails to stop, refer to "15.11 Actions in Response to Failure to Stop an Instance".

3. Recover the failed disk

   Replace the disk, and then recover the volume configuration information.

4. Create a backup data storage destination

   Create a backup data storage destination.

In [Properties] in Windows(R) Explorer, set appropriate permissions so that only the instance administrator can access the backup data storage destination directory. (Refer to [Help and Support] in Windows(R) for information on [Properties].)

Refer to "3.2.2 Using Server Commands" for details.

5. Start the instance

   Start the instance. Refer to "2.1.2 Using Commands" for information on how to start an instance.

6. Run backup

   Use the pgx_dmpall command to back up the database cluster.

   Specify the following value in the pgx_dmpall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

   Example

```
> pgx_dmpall -D D:\database\inst1
```

7. Resume applications

   Resume applications that are using the database.

See

- Refer to "pgx_rcvall" and "pgx_dmpall" in the Reference for information on the pgx_rcvall command and pgx_dmpall command.
- Refer to "Write Ahead Log" under "Server Administration" in the PostgreSQL Documentation for information on archive_command.
- Refer to "B.1 WAL Mirroring Control Functions" for information on pgx_resume_wal_multiplexing.

15.2 Recovering from Data Corruption

If data in a disk is logically corrupted and the database does not operate properly, you can recover the database cluster to its state at the time of backup.

There are two methods of recovery:

- 15.2.1 Using WebAdmin
- 15.2.2 Using the pgx_rcvall Command

Note

- Back up the database cluster after recovering it. Backup deletes obsolete archive logs (transaction logs copied to the backup data storage destination), freeing up disk space and reducing the recovery time.
- If you recover data to a point in the past, a new time series (database update history) will start from that recovery point. When recovery is complete, the recovery point is the latest point in the new time series. When you subsequently recover data to the latest state, the database update is re-executed on the new time series.

15.2.1 Using WebAdmin

If using WebAdmin, recover the data to the point immediately prior to data corruption by using the backup data.

Refer to "15.1.1 Using WebAdmin" for details.

15.2.2 Using the pgx_rcvall Command

Recover the database cluster by specifying in the pgx_rcvall command the date and time of the backup you want to read from. Then re-execute the transaction as required to recover the data.

Follow the procedure below to recover the data storage disk.

1. Stop applications

   Stop applications that are using the database.

2. Stop the instance

   Stop the instance. Refer to "2.1.2 Using Commands" for information on how to stop an instance.

   If the instance fails to stop, refer to "15.11 Actions in Response to Failure to Stop an Instance".

3. Confirm the backup date and time

   Pinpoint a date and time prior to the data corruption based on the content of the job log or event log.

4. Recover the keystore, and enable automatic opening of the keystore

   When the data in the database has been encrypted, restore the keystore to its state at the time of the database backup. Configure automatic opening of the keystore as necessary.

5. Recover the database cluster

   Use the pgx_rcvall command to recover the database cluster.

   Specify the following values in the pgx_rcvall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
   - Specify the backup storage directory in the -B option.
   - Specify the recovery date and time in the -e option.

   Example

   In the following examples, "March 20, 2022 10:00:00" is specified as the recovery time.

   ```
   > pgx_rcvall -D D:\database\inst1 -B E:\backup\inst1 -e "2022-03-20 10:00:00"
   ```

   Note

   If recovery fails, remove the cause of the error in accordance with the displayed error message and then re-execute the pgx_rcvall command.

   If the message "pgx_rcvall: an error occurred during recovery" is displayed, then the log recorded when recovery was executed is output after this message. The cause of the error is output in around the last fifteen lines of the log, so remove the cause of the error in accordance with the message and then re-execute the pgx_rcvall command.

   The following message displayed during recovery is output as part of normal operation of pgx_rcvall command (therefore the user does not need not be concerned).

   ```
   FATAL:  the database system is starting up
   ```

6. Start the instance

   Start the instance. Refer to "2.1.2 Using Commands" for information on how to start an instance.

   If necessary, re-execute transaction processing from the specified recovery time, and then resume database operations.

7. Resume applications

   Resume applications that are using the database.

See

Refer to "pgx_rcvall" in the Reference for information on the pgx_rcvall command.

15.3 Recovering from an Incorrect User Operation

This section describes how to recover database clusters when data has been corrupted due to erroneous user operations.

There are two methods of recovery:

- 15.3.1 Using WebAdmin
- 15.3.2 Using the pgx_rcvall Command

Note

- Back up the database cluster after recovering it. Backup deletes obsolete archive logs (transaction logs copied to the backup data storage destination), freeing up disk space and reducing the recovery time.
- If you recover data to a point in the past, a new time series (database update history) will start from that recovery point. When recovery is complete, the recovery point is the latest point in the new time series. When you subsequently recover data to the latest state, the database update is re-executed on the new time series.
- An effective restore point is one created on a time series for which you have made a backup. That is, if you recover data to a point in the past, you cannot use any restore points set after that recovery point. Therefore, once you manage to recover your target past data, make a backup.

15.3.1 Using WebAdmin

You can use WebAdmin to recover data to a backup point.

Note

Recovery operation cannot be performed on an instance that is part of a streaming replication cluster in standby mode.

If disk failure occurs on a standby instance, it may be necessary to delete and re-create the instance.

Recovery operation can be performed on an instance that is part of a streaming replication cluster in "Master" mode. If a recovery operation is performed on a master instance, it will break the replication cluster and streaming replication will stop between the master instance and all its standby instances. In such an event, the standby instances can be promoted to standalone instances or can be deleted and re-created.

Follow the procedure below to recover the data in the data storage disk.

1. Stop applications

   Stop applications that are using the database.

2. Stop the instance

   Stop the instance. Refer to "2.1.1 Using WebAdmin" for information on how to stop an instance.

3. Recover the keystore, and enable automatic opening of the keystore

   Do the following if the data in the database has been encrypted:

   - Restore the keystore to its state at the time of the database backup.
   - Enable automatic opening of the keystore.

4. Recover the database cluster

   Log in to WebAdmin, and in the [Instances] tab, select the instance to be recovered and click .

5. Recover to the backup point

   In the [Restore Instance] dialog box, click [Yes].

   Recovery is performed. An instance is automatically started when recovery is successful.

6. Resume database operations

   If necessary, re-execute transaction processing from the backup point to when an erroneous operation was performed, and then resume database operations.

15.3.2 Using the pgx_rcvall Command

The pgx_rcvall command recovers database clusters to the restore point created with the server command. Refer to "Setting a restore point" in "3.2.2 Using Server Commands" for information on how to create a restore point.

Follow the procedure below to recover the data in the data storage disk.

1. Stop applications

   Stop applications that are using the database.

2. Stop the instance

   Stop the instance. Refer to "2.1.2 Using Commands" for information on how to stop an instance.

   If the instance fails to stop, refer to "15.11 Actions in Response to Failure to Stop an Instance".

3. Confirm the restore point

   Use a restore point recorded in an arbitrary file, as explained in ""3.2.2 Using Server Commands", to determine a restore point prior to the erroneous operation.

4. Recover the keystore, and enable automatic opening of the keystore

   When the data in the database has been encrypted, restore the keystore to its state at the time of the database backup. Configure automatic opening of the keystore as necessary.

5. Recover the database cluster

   Use the pgx_rcvall command to recover the database cluster.

   Specify the following values in the pgx_rcvall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
   - Specify the backup data storage destination in the -B option.
   - The -n option recovers the data to the specified restore point.

   Example

   The following example executes the pgx_rcvall command with the restore point "batch_20220303_1".

```
> pgx_rcvall -D D:\database\inst1 -B E:\backup\inst1 -n batch_20220303_1
```

Note

If recovery fails, remove the cause of the error in accordance with the displayed error message and then re-execute the pgx_rcvall command.

If the message "pgx_rcvall: an error occurred during recovery" is displayed, then the log recorded when recovery was executed is output after this message. The cause of the error is output in around the last fifteen lines of the log, so remove the cause of the error in accordance with the message and then re-execute the pgx_rcvall command.

The following message displayed during recovery is output as part of normal operation of pgx_rcvall (therefore the user does not need not be concerned).

```
FATAL: the database system is starting up
```

6. Start the instance

   Start the instance.

   Refer to "2.1.2 Using Commands" for information on how to start an instance.

7. Restart operation of the database

   If necessary, re-execute transaction processing from the specified recovery time to the point when an erroneous operation was performed, and then resume database operations.

See

Refer to "pgx_rcvall" in the Reference for information on the pgx_rcvall command.

15.4 Actions in Response to an Application Error

If there is a connection from a client that has been in the waiting state for an extended period, you can minimize performance degradation of the database by closing the problematic connection.

The following methods are available for identifying a connection to be closed:

- view(pg_stat_activity) (refer to "15.4.1 When using the view (pg_stat_activity)")
- pgAdmin (refer to "15.4.2 Using pgAdmin")

Use the system management function (pg_terminate_backend) to disconnect connections.

15.4.1 When using the view (pg_stat_activity)

When using the view (pg_stat_activity), follow the procedure below to close a connection.

1. Use psql command to connect to the postgres database.

```
> psql postgres
psql (<x>) (*1)
Type "help" for help.
```

   *1: *<x>* indicates the PostgreSQL version on which this product is based.

2. Close connections from clients that have been in the waiting state for an extended period.

   Use pg_terminate_backend() to close connections that have been trying to connect for an extended period.

   However, when considering continued compatibility of applications, do not reference or use system catalogs and functions directly in SQL statements. Refer to " Notes on Application Compatibility" in the Application Development Guide for details.

   Example

   The following example closes connections where the client has been in the waiting state for at least 60 minutes.

```
select pid,usename,application_name,client_addr,pg_terminate_backend(pid) from pg_stat_activity
where backend_type = 'client backend' and state='idle in transaction' and current_timestamp >
cast(query_start + interval '60 minutes' as timestamp);
-[ RECORD 1 ]--------+---------------
pid                  | 4684
username             | fsepuser
application_name     | apl1
client_addr          | 192.11.11.1
pg_terminate_backend | t
```

See

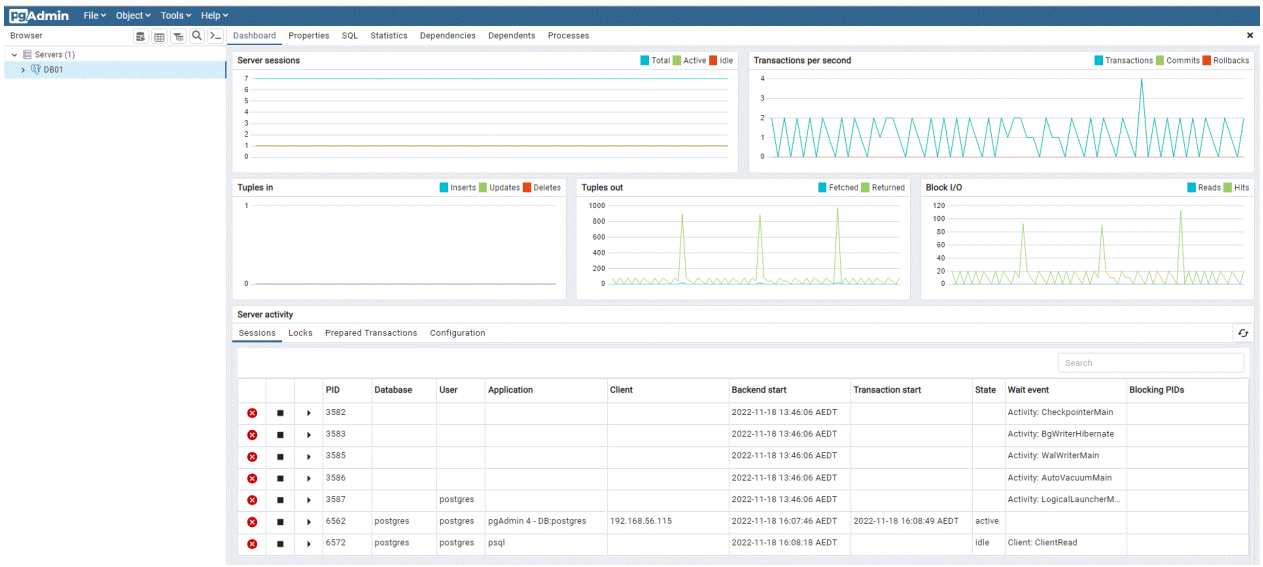
- Refer to "System Administration Functions" under "The SQL Language" in the PostgreSQL Documentation for information on pg_terminate_backend.
- Refer to "Notes on Application Compatibility" in the Application Development Guide for information on how to maintain application compatibility.

15.4.2 Using pgAdmin

If using pgAdmin, follow the procedure below to close connections.

1. In the [Browser] pane, click the relevant database server.
2. Close the client connections that have been in a wait state for an extended period.

   Click the [Dashboard] tab. In the [Server activity] section, select the connections that have been in an "idle" or "idle in transaction" state for an extended period. For each of these connections, click to close the session.



15.5 Actions in Response to an Access Error

If access is denied, grant privileges allowing the instance administrator to operate the following directories, and then re-execute the operation. Also, refer to the event log and the server log, and confirm that the file system has not been mounted as read-only due to a disk error. If the file system has been mounted as read-only, mount it properly and then re-execute the operation.

- Data storage destination
- Tablespace storage destination
- Transaction log storage destination
- Backup data storage destination

See

Refer to "Preparing Directories to Deploy Resources" under "Setup" in the Installation and Setup Guide for Server for information on the privileges required for the directory.

15.6 Actions in Response to Insufficient Space on the Data Storage Destination

If the data storage destination runs out of space, check if the disk contains any unnecessary files and delete them so that operations can continue.

If deleting unnecessary files does not solve the problem, you must migrate data to a disk with larger capacity.

There are two methods of migrating data:

- 15.6.1 Using a Tablespace
- 15.6.2 Replacing the Disk with a Larger Capacity Disk

15.6.1 Using a Tablespace

Fujitsu Enterprise Postgres enables you to use a tablespace to change the storage destination of database objects, such as tables and indexes, to a different disk.

The procedure is as follows:

1. Create a tablespace

   Use the CREATE TABLESPACE command to create a new tablespace in the prepared disk.

2. Modify the tablespace

   Use the ALTER TABLE command to modify tables for the newly defined tablespace.

See

Refer to "SQL Commands" under "Reference" in the PostgreSQL Documentation for information on the CREATE TABLESPACE command and ALTER TABLE command.

15.6.2 Replacing the Disk with a Larger Capacity Disk

Before replacing the disk with a larger capacity disk, migrate resources at the data storage destination using the backup and recovery features.

There are two methods of performing backup and recovery:

- 15.6.2.1 Using WebAdmin
- 15.6.2.2 Using Server Commands

The following sections describe procedures that use each of these methods to replace the disk and migrate resources at the data storage destination.

Note

- Before replacing the disk, stop applications and instances that are using the database.
- It is recommended that you back up the database cluster following recovery. Backup deletes obsolete archive logs (transaction logs copied to the backup data storage destination), freeing up disk space and reducing the recovery time.

15.6.2.1 Using WebAdmin

Follow the procedure below to replace the disk and migrate resources at the data storage destination by using WebAdmin.

1. Back up files

   If the disk at the data storage destination contains any required files, back up the files. It is not necessary to back up the data storage destination.

2. Stop applications

   Stop applications that are using the database.

3. Back up the database cluster

   Back up the latest resources at the data storage destination. Refer to "3.2.1 Using WebAdmin" for details.

4. Stop the instance

   Stop the instance. Refer to "2.1.1 Using WebAdmin" for information on how to stop an instance.

5. Replace with a larger capacity disk

   Replace the disk. Then, recover the volume configuration information.

6. Recover the database cluster

   Log in to WebAdmin, and perform recovery operations. Refer to steps 4 ("Create a tablespace directory ") to 7 ("Run recovery") under "If failure occurred in the data storage disk or the transaction log storage disk" in "15.1.1 Using WebAdmin" for information on the procedure. An instance is automatically started when recovery is successful.

7. Resume applications

   Resume applications that are using the database.

8. Restore the files

   Restore the files backed up in step 1.

15.6.2.2 Using Server Commands

Follow the procedure below to replace the disk and migrate resources at the data storage destination by using server commands.

1. Back up files

   If the disk at the data storage destination contains any required files, back up the files. It is not necessary to back up the data storage destination.

2. Stop applications

   Stop applications that are using the database.

3. Back up the database cluster

   Back up the latest resources at the data storage destination. Refer to "3.2.2 Using Server Commands" for details.

4. Stop the instance

   After backup is complete, stop the instance. Refer to "2.1.2 Using Commands" for information on how to stop an instance.

   If the instance fails to stop, refer to "15.11 Actions in Response to Failure to Stop an Instance".

5. Replace with a larger capacity disk

   Replace the disk. Then, recover the volume configuration information.

6. Create a data storage destination

   Create a data storage destination. If a tablespace was defined, also create a directory for it.

   In [Properties] in Windows(R) Explorer, set appropriate permissions so that only the instance administrator can access the data storage destination directory. (Refer to [Help and Support] in Windows(R) for information on [Properties].)

7. Recover the keystore, and enable automatic opening of the keystore

   When the data in the database has been encrypted, restore the keystore to its state at the time of the database backup. Configure automatic opening of the keystore as necessary.

8. Recover the database cluster

   Use the pgx_rcvall command to recover the database cluster.

- Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
- Specify the backup storage directory in the -B option.

Example

```
> pgx_rcvall -D D:\database\inst1 -B E:\backup\inst1
```

Note

If recovery fails, remove the cause of the error in accordance with the displayed error message and then re-execute the pgx_rcvall command.

If the message "pgx_rcvall: an error occurred during recovery" is displayed, then the log recorded when recovery was executed is output after this message. The cause of the error is output in around the last fifteen lines of the log, so remove the cause of the error in accordance with the message and then re-execute the pgx_rcvall command.

The following message displayed during recovery is output as part of normal operation of pgx_rcvall (therefore the user does not need not be concerned).

```
FATAL:  the database system is starting up
```

See

Refer to "pgx_rcvall" in the Reference for information on the pgx_rcvall command.

9. Start the instance

   Start the instance.

   Refer to "2.1.2 Using Commands" for information on how to start an instance.

10. Resume applications

    Resume applications that are using the database.

11. Restore files

    Restore the files backed up in step 1.

15.7 Actions in Response to Insufficient Space on the Backup Data Storage Destination

If space runs out on the backup data storage destination, check if the disk contains any unnecessary files and delete them, and then make a backup as required.

If deleting unnecessary files does not solve the problem, take the following action:

- 15.7.1 Temporarily Saving Backup Data
- 15.7.2 Replacing the Disk with a Larger Capacity Disk

15.7.1 Temporarily Saving Backup Data

This method involves temporarily moving backup data to a different directory, saving it there, and securing disk space on the backup data storage destination so that a backup can be made normally.

Use this method if you need time to prepare a larger capacity disk.

If space runs out on the backup data storage destination, archive logs can no longer be stored in the backup data storage destination. As a result, transaction logs continue to accumulate in the data storage destination or the transaction log storage destination.

If action is not taken soon, the transaction log storage destination will become full, and operations may not be able to continue.

To prevent this, secure space in the backup data storage destination, so that archive logs can be stored.

There are two methods of taking action:

- 15.7.1.1 Using WebAdmin
- 15.7.1.2 Using Server Commands

15.7.1.1 Using WebAdmin

Follow the procedure below to recover the backup data storage disk.

1. Temporarily save backup data

   Move backup data to a different directory and temporarily save it, and secure space in the backup data storage destination directory.

   The reason for saving the backup data is so that the data in the data storage destination can be recovered even if it is corrupted before you perform recovery. If there is no disk at the save destination and you consider that there is no risk of corruption at the data storage destination, delete the backup data.

   The following example saves backup data from the backup data storage destination directory (E:\backup\inst1) under F:\mnt\usb \backup.

   Example

   ```
   > mkdir F:\mnt\usb\backup
   > move E:\backup\inst1\* F:\mnt\usb\backup
   ```

   Note: Place the temporary backup destination directory in a location where it will not impact on operating system resources or Fujitsu Enterprise Postgres resources.

2. Back up the database cluster

   Back up the latest resources at the data storage destination. Refer to "3.2.1 Using WebAdmin" for details.

3. Delete temporarily saved backup data

   If backup completes normally, the temporarily saved backup data becomes unnecessary and is deleted.

   The following example deletes backup data that was temporarily saved in F:\mnt\usb.

   Example

   ```
   > rmdir /S /Q F:\mnt\usb\backup
   ```

15.7.1.2 Using Server Commands

The following describes the procedure for recovering the backup storage disk.

There are two methods of taking action:

- Performing recovery while the instance is active
- Stopping the instance before performing recovery

The following table shows the different steps to be performed depending on whether you stop the instance.

No	Step	Instance stopped	
		No	Yes
1	Stop transaction log mirroring	Y	N
2	Stop output of archive logs	Y	N
3	Stop applications	N	Y
4	Stop the instance	N	Y

No	Step	Instance stopped	
		No	Yes
5	Temporarily save backup data	Y	Y
6	Resume output of archive logs	Y	N
7	Resume transaction log mirroring	Y	N
8	Start an instance	N	Y
9	Run backup	Y	Y
10	Resume applications	N	Y
11	Delete temporarily saved backup data	Y	Y

Y: Required
N: Not required

The procedure is as follows:

Performing recovery while the instance is active

1. Stop transaction log mirroring

   Stop transaction log mirroring.

```
postgres=# SELECT pgx_pause_wal_multiplexing();
LOG:  multiplexing of transaction log files has been stopped
pgx_pause_wal_multiplexing
---------------------------

(1 row)
```

2. Stop output of archive logs

   Transaction logs may accumulate during replacement of backup storage disk, and if the data storage disk or the transaction log storage disk becomes full, there is a risk that operations may not be able to continue.

   To prevent this, use the following methods to stop output of archive logs.

   - Changing the archive_command parameter

     Specify a command that will surely complete normally, so that archive logs will be regarded as having been output.

     If you specify echo, a message is output to the server log, so it may be used as a reference when you conduct investigations.

   - Reloading the configuration file

     Run the pg_ctl reload command or the pg_reload_conf SQL function.

   If you simply want to stop output of errors without the risk that operations will not be able to continue, specify an empty string ('') in archive_command and reload the configuration file.

3. Temporarily save backup data

   Move backup data to a different directory and temporarily save it, and secure space in the backup data storage destination directory.

   The reason for saving the backup data is so that the data in the data storage destination can be recovered even if it is corrupted before you perform the next step. If there is no disk at the save destination and you consider that there is no risk of corruption at the data storage destination, delete the backup data.

   The following example saves backup data from the backup data storage destination directory (E:\backup\inst1) under F:\mnt\usb \backup.

Example

```
> mkdir F:\mnt\usb\backup
> move E:\backup\inst1\* F:\mnt\usb\backup
```

Note: Place the temporary backup destination directory in a location where it will not impact on operating system resources or Fujitsu Enterprise Postgres resources.

4. Resume output of archive logs

   Return the archive_command setting to its original value, and reload the configuration file.

5. Resume transaction log mirroring

   Execute the pgx_resume_wal_multiplexing SQL function.

   Example

   ```
   SELECT pgx_resume_wal_multiplexing()
   ```

6. Run backup

   Use the pgx_dmpall command to back up the database cluster.

   Specify the following option in the pgx_dmpall command:

   - Specify the directory of the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

     Example

     ```
     > pgx_dmpall -D D:\database\inst1
     ```

7. Delete temporarily saved backup data

   If backup completes normally, the temporarily saved backup data becomes unnecessary and is deleted.

   The following example deletes backup data that was temporarily saved in F:\mnt\usb.

   Example

   ```
   > rmdir /S /Q F:\mnt\usb\backup
   ```

If an instance has been stopped

1. Stop applications

   Stop applications that are using the database.

2. Stop the instance

   Stop the instance. Refer to "2.1.2 Using Commands" for details.

   If the instance fails to stop, refer to "15.11 Actions in Response to Failure to Stop an Instance".

3. Temporarily save backup data

   Move backup data to a different directory and temporarily save it, and secure space in the backup data storage destination directory.

   The reason for saving the backup data is so that the data in the data storage destination can be recovered even if it is corrupted before you perform recovery. If there is no disk at the save destination and you consider that there is no risk of corruption at the data storage destination, delete the backup data.

   The following example saves backup data from the backup data storage destination directory (E:\backup\inst1) under F:\mnt\usb\backup.

   Example

   ```
   > mkdir F:\mnt\usb\backup
   > move E:\backup\inst1\* F:\mnt\usb\backup
   ```

Note: Place the temporary backup destination directory in a location where it will not impact on operating system resources or Fujitsu Enterprise Postgres resources.

4. Start the instance

   Start the instance. Refer to "2.1.2 Using Commands" for information on how to start an instance.

5. Run backup

   Use the pgx_dmpall command to back up the database cluster.

   Specify the following value in the pgx_dmpall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

     Example

     ```
     > pgx_dmpall -D D:\database\inst1
     ```

6. Resume applications

   Resume applications that are using the database.

7. Delete temporarily saved backup data

   If backup completes normally, the temporarily saved backup data becomes unnecessary and is deleted.

   The following example deletes backup data that was temporarily saved in F:\mnt\usb.

   Example

   ```
   > rmdir /S /Q F:\mnt\usb\backup
   ```

See

- Refer to "pgx_rcvall" and "pgx_dmpall" in the Reference for information on the pgx_rcvall command and pgx_dmpall command.
- Refer to "Write Ahead Log" under "Server Administration" in the PostgreSQL Documentation for information on archive_command.
- Refer to "B.1 WAL Mirroring Control Functions" for information on the pgx_is_wal_multiplexing_paused and pgx_resume_wal_multiplexing.

15.7.2 Replacing the Disk with a Larger Capacity Disk

This method involves replacing the disk at the backup data storage destination with a larger capacity disk, so that it does not run out of free space again. After replacing the disk, back up data to obtain a proper backup.

There are two methods of performing backup:

- 15.7.2.1 Using WebAdmin
- 15.7.2.2 Using Server Commands

Note

Before replacing the disk, stop applications that are using the database.

15.7.2.1 Using WebAdmin

Follow the procedure below to recover the backup storage disk.

1. Back up files

   If the disk at the backup data storage destination contains any required files, back up the files. It is not necessary to back up the backup data storage destination.

2. Temporarily save backup data

   Save the backup data to a different directory.

   The reason for saving the backup data is so that the data in the data storage destination can be recovered even if it is corrupted before you perform the next step. If there is no disk at the save destination and you consider that there is no risk of corruption at the data storage destination, delete the backup data.

   The following example saves backup data from the backup data storage destination directory (E:\backup\inst1) under F:\mnt\usb \backup.

   Example

   ```
   > mkdir F:\mnt\usb\backup
   > move E:\backup\inst1\* F:\mnt\usb\backup
   ```

   Note: Place the temporary backup destination directory in a location where it will not impact on operating system resources or Fujitsu Enterprise Postgres resources.

3. Replace with a larger capacity disk

   Replace the disk. Then, recover the volume configuration information.

4. Run backup

   Log in to WebAdmin, and perform recovery operations. Refer to steps 2 ("Recover the backup data") and 3 ("Run backup") under "If failure occurred on the backup storage disk" in "15.1.1 Using WebAdmin".

5. Restore files

   Restore the files backed up in step 1.

6. Delete temporarily saved backup data

   If backup completes normally, the temporarily saved backup data becomes unnecessary and is deleted.

   The following example deletes backup data that was temporarily saved in F:\mnt\usb.

   Example

   ```
   > rmdir /S /Q F:\mnt\usb\backup
   ```

15.7.2.2 Using Server Commands

The procedure for recovering the backup data storage disk is described below.

There are two methods of taking action:

- Performing recovery while the instance is active
- Stopping the instance before performing recovery

The following table shows the different steps to be performed depending on whether you stop the instance.

No	Step	Instance stopped	
		No	Yes
1	Back up files	Y	Y
2	Temporarily save backup data	Y	Y
3	Confirm that transaction log mirroring has stopped	Y	N
4	Stop output of archive logs	Y	N

No	Step	Instance stopped	
		No	Yes
5	Stop applications	N	Y
6	Stop the instance	N	Y
7	Replace with a larger capacity disk	Y	Y
8	Create a backup storage directory	Y	Y
9	Resume output of archive logs	Y	N
10	Resume transaction log mirroring	Y	N
11	Start the instance	N	Y
12	Run backup	Y	Y
13	Resume applications	N	Y
14	Restore files	Y	Y
15	Delete temporarily saved backup data	Y	Y

Y: Required
N: Not required

The procedure is as follows:

If an instance has not been stopped

1. Back up files

   If the disk at the backup data storage destination contains any required files, back up the files. It is not necessary to back up the backup data storage destination.

2. Temporarily save backup data

   Save the backup data to a different directory.

   The reason for saving the backup data is so that the data in the data storage destination can be recovered even if it is corrupted before you perform the next step. If there is no disk at the save destination and you consider that there is no risk of corruption at the data storage destination, delete the backup data.

   The following example saves backup data from the backup data storage destination directory (E:\backup\inst1) under F:\mnt\usb\backup.

   #### Example

   ```
   > mkdir F:\mnt\usb\backup
   > move E:\backup\inst1\* F:\mnt\usb\backup
   ```

3. Confirm that transaction log mirroring has stopped

   Use the following SQL function to confirm that transaction log mirroring has stopped.

   ```
    postgres=# SELECT pgx_is_wal_multiplexing_paused();
    pgx_is_wal_multiplexing_paused
   -------------------
    t
   (1 row)
   ```

   If transaction log mirroring has not stopped, then stop it using the following SQL function.

   ```
   postgres=# SELECT pgx_pause_wal_multiplexing();
   LOG:  multiplexing of transaction log files has been stopped
   pgx_pause_wal_multiplexing
   ---------------------------
   ```

```
(1 row)
```

4. Stop output of archive logs

   Transaction logs may accumulate during replacement of backup storage disk, and if the data storage destination disk or the transaction log storage destination disk becomes full, there is a risk that operations may not be able to continue.

   To prevent this, use the following methods to stop output of archive logs.

   - Changing the archive_command parameter

     Specify a command that will surely complete normally, so that archive logs will be regarded as having been output.

     If you specify echo, a message is output to the server log, so it may be used as a reference when you conduct investigations.

   - Reloading the configuration file

     Run the pg_ctl reload command or the pg_reload_conf SQL function.

   If you simply want to stop output of errors without the risk that operations will not be able to continue, specify an empty string ('') in archive_command and reload the configuration file.

5. Replace with a larger capacity disk

   Replace the disk. Then, recover the volume configuration information.

6. Create a backup data storage destination

   Create a backup data storage destination.

   In [Properties] in Windows(R) Explorer, set appropriate permissions so that only the instance administrator can access the backup data storage destination directory. (Refer to [Help and Support] in Windows(R) for information on [Properties].)

   Refer to "3.2.2 Using Server Commands" for details.

7. Resume output of archive logs

   Return the archive_command setting to its original value, and reload the configuration file.

8. Resume transaction log mirroring

   Execute the pgx_resume_wal_multiplexing SQL function.

   Example

   ```
   SELECT pgx_resume_wal_multiplexing()
   ```

9. Run backup

   Use the pgx_dmpall command to back up the database cluster.

   Specify the following value in the pgx_dmpall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

   Example

   ```
   > pgx_dmpall -D D:\database\inst1
   ```

10. Restore files

    Restore the files backed up in step 1.

11. Delete temporarily saved backup data

    If backup completes normally, the temporarily saved backup data becomes unnecessary and is deleted.

    The following example deletes backup data that was temporarily saved in F:\mnt\usb.

Example

```
> rmdir /S /Q F:\mnt\usb\backup
```

If an instance has been stopped

1. Back up files

   If the disk at the backup data storage destination contains any required files, back up the files. It is not necessary to back up the backup data storage destination.

2. Temporarily save backup data

   Save the backup data to a different directory.

   The reason for saving the backup data is so that the data in the data storage destination can be recovered even if it is corrupted before you perform the next step. If there is no disk at the save destination and you consider that there is no risk of corruption at the data storage destination, delete the backup data.

   The following example saves backup data from the backup data storage destination directory (E:\backup\inst1) under F:\mnt\usb\backup.

   Example

   ```
   > mkdir F:\mnt\usb\backup
   > move E:\backup\inst1\* F:\mnt\usb\backup
   ```

   Note: Place the temporary backup destination directory in a location where it will not impact on operating system resources or Fujitsu Enterprise Postgres resources.

3. Stop applications

   Stop applications that are using the database.

4. Stop the instance

   Stop the instance. Refer to "2.1.2 Using Commands" for information on how to stop an instance.

   If the instance fails to stop, refer to "15.11 Actions in Response to Failure to Stop an Instance".

5. Replace with a larger capacity disk

   Replace the disk. Then, recover the volume configuration information.

6. Create a backup data storage destination

   Create a backup data storage destination.

   In [Properties] in Windows(R) Explorer, set appropriate permissions so that only the instance administrator can access the backup data storage destination directory. (Refer to [Help and Support] in Windows(R) for information on [Properties].)

   Refer to "3.2.2 Using Server Commands" for details.

7. Start the instance

   Start the instance. Refer to "2.1.2 Using Commands" for information on how to start an instance.

8. Run backup

   Use the pgx_dmpall command to back up the database cluster.

   Specify the following value in the pgx_dmpall command:

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.

   Example

   ```
   > pgx_dmpall -D D:\database\inst1
   ```

9. Resume applications

   Resume applications that are using the database.

10. Restore files

    Restore the files backed up in step 1.

11. Delete temporarily saved backup data

    If backup completes normally, the temporarily saved backup data becomes unnecessary and is deleted.

    The following example deletes backup data that was temporarily saved in F:\mnt\usb.

    Example

    ```
    > rmdir /S /Q F:\mnt\usb\backup
    ```

See

- Refer to "pgx_rcvall" and "pgx_dmpall" in the Reference for information on the pgx_rcvall command and pgx_dmpall command.
- Refer to "Write Ahead Log" under "Server Administration" in the PostgreSQL Documentation for information on archive_command.
- Refer to "B.1 WAL Mirroring Control Functions" for information on the pgx_is_wal_multiplexing_paused and pgx_resume_wal_multiplexing.

15.8 Actions in Response to Insufficient Space on the Transaction Log Storage Destination

If the transaction log storage destination runs out of space, check if the disk contains any unnecessary files and delete them so that operations can continue.

If deleting unnecessary files does not solve the problem, you must migrate data to a disk with larger capacity.

15.8.1 Replacing the Disk with a Larger Capacity Disk

Before replacing the disk with a larger capacity disk, migrate resources at the transaction log storage destination using the backup and recovery features.

There are two methods of performing backup and recovery:

- 15.8.1.1 Using WebAdmin
- 15.8.1.2 Using Server Commands

The following sections describe procedures that use each of these methods to replace the disk and migrate resources at the transaction log storage destination.

Note

- Before replacing the disk, stop applications that are using the database.
- It is recommended that you back up the database cluster following recovery. Backup deletes obsolete archive logs (transaction logs copied to the backup data storage destination), freeing up disk space and reducing the recovery time.

15.8.1.1 Using WebAdmin

Follow the procedure below to replace the disk and migrate resources at the transaction log storage destination by using WebAdmin.

1. Back up files

   If the disk at the transaction log storage destination contains any required files, back up the files. It is not necessary to back up the transaction log storage destination.

2. Back up the database cluster

   Back up the latest data storage destination resources and transaction log storage destination resources (refer to "3.2.1 Using WebAdmin" for details).

3. Stop applications

   Stop applications that are using the database.

4. Stop the instance

   Stop the instance. Refer to "2.1.1 Using WebAdmin" for information on how to stop an instance. WebAdmin automatically stops instances if recovery of the database cluster is performed without stopping the instance.

5. Replace with a larger capacity disk

   Replace the disk. Then, recover the volume configuration information.

6. Create a tablespace directory

   If a tablespace was defined after backing up, create a directory for it.

7. Recover the keystore, and enable automatic opening of the keystore

   Do the following if the data in the database has been encrypted:

   - Restore the keystore to its state at the time of the database backup.
   - Enable automatic opening of the keystore.

8. Recover the database cluster

   Log in to WebAdmin, and perform recovery operations. Refer to steps 4 ("Create a tablespace directory ") to 7 ("Run Recovery") under " If failure occurred in the data storage disk or the transaction log storage disk " in "15.1.1 Using WebAdmin" for information on the procedure. An instance is automatically started when recovery is successful.

9. Resume applications

   Resume applications that are using the database.

10. Restore files

    Restore the files backed up in step 1.

15.8.1.2 Using Server Commands

Follow the procedure below to replace the disk and migrate resources at the transaction log storage destination by using server commands.

1. Back up files

   If the disk at the transaction log storage destination contains any required files, back up the files. It is not necessary to back up the transaction log storage destination.

2. Back up the database cluster

   Use server commands to back up the latest data storage destination resources and transaction log storage destination resources. Refer to "3.2.2 Using Server Commands" for information on how to perform backup.

3. Stop applications

   Stop applications that are using the database.

4. Stop the instance

   After backup is complete, stop the instance. Refer to "2.1.2 Using Commands" for information on how to stop an instance.

   If the instance fails to stop, refer to "15.11 Actions in Response to Failure to Stop an Instance".

5. Replace with a larger capacity disk

   Replace the disk. Then, recover the volume configuration information.

6. Create a transaction log storage destination

   Create a transaction log storage destination. If a tablespace was defined, also create a directory for it.

   In [Properties] in Windows(R) Explorer, set appropriate permissions so that only the instance administrator can access the transaction log destination directory. (Refer to [Help and Support] in Windows(R) for information on [Properties].)

7. Recover the keystore, and enable automatic opening of the keystore

   When the data in the database has been encrypted, restore the keystore to its state at the time of the database backup. Configure automatic opening of the keystore as necessary.

8. Recover the database cluster

   Use the pgx_rcvall command to recover the database cluster.

   - Specify the data storage destination in the -D option. If the -D option is omitted, the value of the PGDATA environment variable is used by default.
   - Specify the backup storage directory in the -B option.

   Example

```
> pgx_rcvall -D D:\database\inst1 -B E:\backup\inst1
```

   Note

   If recovery fails, remove the cause of the error in accordance with the displayed error message and then re-execute the pgx_rcvall command.

   If the message "pgx_rcvall: an error occurred during recovery" is displayed, then the log recorded when recovery was executed is output after this message. The cause of the error is output in around the last fifteen lines of the log, so remove the cause of the error in accordance with the message and then re-execute the pgx_rcvall command.

   The following message displayed during recovery is output as part of normal operation of pgx_rcvall command (therefore the user does not need not be concerned).

```
FATAL:  the database system is starting up
```

   See

   Refer to "pgx_rcvall" in the Reference for information on the pgx_rcvall command.

9. Start the instance

   Start the instance.

   Refer to "2.1.2 Using Commands" for information on how to start an instance.

10. Resume applications

    Resume applications that are using the database.

11. Restore files

    Restore the files backed up in step 1.

15.9 Errors in More Than One Storage Disk

If an error occurs in the storage destination disks or resources are corrupted, determine the cause of the error from event logs and server logs and remove the cause.

If errors occur in either of the following combinations, you cannot recover the database.

Recreate the instance, and rebuild the runtime environment.

Data storage disk	Transaction log storage disk	Backup data storage disk
Error	-	Error
-	Error	Error

See

Refer to "Setup" in the Installation and Setup Guide for Server for information on how to create an instance and build the runtime environment.

15.10 Actions in Response to Instance Startup Failure

If an instance fails to start, refer to the event log and the server log, and determine the cause of the failure.

If using WebAdmin, remove the cause of the error. Then, click [Solution] and [Recheck the status] and confirm that the instance is in the normal state.

The following sections describe common causes of errors and the actions to take.

15.10.1 Errors in the Configuration File

If you have directly edited the configuration file using a text editor or changed the settings using WebAdmin, refer to the event log and the server log, confirm that no messages relating to the files below have been output.

- postgresql.conf
- pg_hba.conf

See

Refer to the following for information on the parameters in the configuration file:

- "Configuring Parameters" in the Installation and Setup Guide for Server
- "Appendix A Parameters"
- "Server Configuration" and "Client Authentication" under "Server Administration" in the PostgreSQL Documentation

15.10.2 Errors Caused by Power Failure or Mounting Issues

If mounting is cancelled after restarting the server, for example, because the disk device for each storage destination disk was not turned on, or because automatic mounting has not been set, then starting an instance will fail.

Refer to "15.14.2 Errors Caused by Power Failure or Mounting Issues", and take actions accordingly.

15.10.3 Other Errors

This section describes the recovery procedure to be used if you cannot take any action or the instance cannot start even after you have referred to the event log and the server log.

There are two methods of recovery:

- 15.10.3.1 Using WebAdmin
- 15.10.3.2 Using Server Commands

Note that recovery will not be possible if there is an error at the backup data storage destination. If the problem cannot be resolved, contact Fujitsu technical support.

15.10.3.1 Using WebAdmin

Follow the procedure below to perform recovery.

1. Delete the data storage destination directory and the transaction log storage destination directory

   Back up the data storage destination directory and the transaction log storage destination directory before deleting them.

2. Reconfirm the status

   Log in to WebAdmin, and in the [Instances] tab, click [Solution] for the error message.

   Click [Recheck the status] to reconfirm the storage destination resources.

3. Run recovery

   Restore the database cluster after WebAdmin detects an error.

   Refer to "15.2.1 Using WebAdmin" for details.

15.10.3.2 Using Server Commands

Follow the procedure below to recover the database.

1. Delete the data storage destination directory and the transaction log storage destination directory

   Save the data storage destination directory and the transaction log storage destination directory, and then delete them.

2. Execute recovery

   Use the pgx_rcvall command to recover the database cluster.

   Refer to "15.2.2 Using the pgx_rcvall Command" for details.

15.11 Actions in Response to Failure to Stop an Instance

If an instance fails to stop, refer to the event log and the server log, and determine the cause of the failure.

If the instance cannot stop despite taking action, perform the following operation to stop the instance.

There are two methods of recovery:

- 15.11.1 Using WebAdmin
- 15.11.2 Using Server Commands

15.11.1 Using WebAdmin

In the [Instances] tab, click and select the Fast stop mode or the Immediate stop mode to stop the instance. Forcibly terminate the server process from WebAdmin if the instance cannot be stopped.

Refer to "2.1.1 Using WebAdmin" for information on the stop modes.

15.11.2 Using Server Commands

There are three methods:

- Stopping the Instance Using the Fast Mode

  If backup is in progress, then terminate it, roll back all executing transactions, forcibly close client connections, and then stop the instance.

- Stopping the Instance Using the Immediate Mode

  Forcibly terminate the instance immediately. A crash recovery is run when the instance is restarted.

- Forcibly Stopping the Server Process

  Reliably stops the server process when the other methods are unsuccessful.

15.11.2.1 Stopping the Instance Using the Fast Mode

Specify "-m fast" in the pg_ctl command to stop the instance.

If the instance fails to stop when you use this method, stop the instance as described in "15.11.2.2 Stopping the Instance Using the Immediate Mode" or "15.11.2.3 Forcibly Stopping the Server Process".

Example

```
> pg_ctl stop -D D:\database\inst1 -m fast
```

15.11.2.2 Stopping the Instance Using the Immediate Mode

Specify "-m immediate " in the pg_ctl command to stop the instance.

If the instance fails to stop when you use this method, stop the instance as described in "15.11.2.3 Forcibly Stopping the Server Process".

Example

```
> pg_ctl stop -D D:\database\inst1 -m immediate
```

15.11.2.3 Forcibly Stopping the Server Process

If both the Fast mode and the Immediate mode fail to stop the instance, use the kill parameter of the pg_ctl command to forcibly stop the server process.

The procedure is as follows:

1. Execute the wmic command to identify the process ID of the server process.

   ```
   c:\>wmic
   wmic:root\cli>process where "name = \"postgres.exe\"" get CommandLine,Name,ProcessId
   CommandLine                                                                    Name
   ProcessId
   :
   "C:\Program Files\Fujitsu\fsepv<x>server64\bin\postgres.exe" -D "D:\database\inst1"
   postgres.exe  896
   :
   ```

   The postgres.exe process ID(896) that indicates the data storage destination directory of the applicable instance in the -D option becomes the server process.

2. Forcibly stop the server process

   As instance manager, forcibly stop the server process using the pg_ctl command.

   ```
   c:\>pg_ctl kill QUIT 896
   ```

15.12 Actions in Response to Failure to Create a Streaming Replication Standby Instance

When creating a streaming replication standby instance using WebAdmin, if the instance creation fails, refer to the event log and the server log, and determine the cause of the failure.

When an error occurs in the creation of the standby instance using WebAdmin, it is unlikely that the partially created standby instance can be resumed to complete the operation.

In such a scenario, fix the cause of the error, delete the partially created standby instance, and then create a new standby instance. This recommendation is based on the following assumptions:

- As the instance is yet to be created completely, there are no applications connecting to the database.
- The standby instance is in error state and is not running.
- There are no backups for the standby instance and as a result, it cannot be recovered.

Refer to "Deleting Instances" in the Installation and Setup Guide for details on how to delete an instance.

15.13 Actions in Response to Error in a Distributed Transaction

If a system failure (such as server failure) occurs in an application that uses distributed transactions (such as .NET TransactionScope), then transactions may be changed to the in-doubt state.

At that point, resources accessed by the transaction will be locked, and rendered unusable by other transactions.

The following describes how to check for in-doubt transactions, and how to resolve them.

How to check for in-doubt transactions

The following shows how to check for them:

If the server fails

1. An in-doubt transaction will have occurred if a message similar to the one below is output to the log when the server is restarted.

   Example

   ```
   LOG: Restoring prepared transaction 2103.
   ```

2. Refer to system view pg_prepared_xacts to obtain information about the prepared transaction.

   If the transaction identifier of the prepared transaction in the list (in the transaction column of pg_prepared_xacts) is the same as the identifier of the in-doubt transaction obtained from the log output when the server was restarted, then that row is the information about the in-doubt transaction.

   Example

   ```
   postgres=# select * from pg_prepared_xacts;
    transaction |   gid     |   prepared |  owner   | database
   -------------+-----------+------------+----------+----------
   2103 | 374cc221-f6dc-4b73-9d62-d4fec9b430cd | 2022-03-06 16:28:48.471+08 | postgres |
   postgres (1 row)
   ```

   Information about the in-doubt transaction is output to the row with the transaction ID 2103 in the transaction column.

If the client fails

If there are no clients connected and there is a prepared transaction in pg_prepared_xacts, then you can determine that the transaction is in the in-doubt state.

If at least one client is connected and there is a prepared transaction in pg_prepared_xacts, you cannot determine whether there is a transaction in the in-doubt state. In this case, use the following query to determine the in-doubt transaction from the acquired database name, user name, the time PREPARE TRANSACTION was executed, and the information about the table name accessed.

```
select gid,x.database,owner,prepared,l.relation::regclass as relation from pg_prepared_xacts x
left join pg_locks l on l.virtualtransaction = '-1/'||x.transaction and l.locktype='relation';
```

If it still cannot be determined from this information, wait a few moments and then check pg_prepared_xacts again.

If there is a transaction that has continued since the last time you checked, then it is likely that it is the one in the in-doubt state.

Point

As you can see from the explanations in this section, there is no one way to definitively determine in-doubt transactions.

Consider collecting other supplementary information (for example, logging on the client) or performing other operations (for example, allocating database users per job).

How to resolve in-doubt transactions

From the system view pg_prepared_xacts mentioned above, obtain the global transaction identifier (in the gid column of pg_prepared_xacts) for the in-doubt transaction, and issue either a ROLLBACK PREPARED statement or COMMIT PREPARED statement to resolve the in-doubt transaction.

Example

- Rolling back in-doubt transactions

```
postgres=# rollback prepared '374cc221-f6dc-4b73-9d62-d4fec9b430cd';
ROLLBACK PREPARED
```

- Committing in-doubt transactions

```
postgres=# commit prepared '374cc221-f6dc-4b73-9d62-d4fec9b430cd';
COMMIT PREPARED
```

15.14 I/O Errors Other than Disk Failure

Even if a disk is not defective, the same input-output error messages, as those generated when the disk is defective, may be output.

A few examples of such errors are given below. The appropriate action for each error is explained respectively.

- 15.14.1 Network Error with an External Disk
- 15.14.2 Errors Caused by Power Failure or Mounting Issues

15.14.1 Network Error with an External Disk

This is an error that occurs in the network path to/from an external disk.

Determine the cause of the error by checking the information in the event log and the server log, the disk access LED, network wiring, and network card status. Take appropriate action to remove the cause of the error, for example, replace problematic devices.

15.14.2 Errors Caused by Power Failure or Mounting Issues

These are errors that occur when the disk device is not turned on, automatic mounting of the disk was not set, or mounting was accidentally cancelled.

In this case, check the information in the event log and the server log, the disk access LED, and whether the disk is mounted correctly. If problems are detected, take appropriate action.

If mounting has been cancelled, it is possible that mounting was accidentally cancelled, or the existing setting (automatic mounting at the time of starting the operating system) has been changed so that mounting is not performed automatically. In this case, set the mounting to be performed automatically.

15.15 Anomaly Detection and Resolution

The following operations performed via the command line interface will result in an anomaly in WebAdmin:

- Changes to the port and backup_destination parameters in postgresql.conf

- Changes to Mirroring Controller configuration of cluster replication added via WebAdmin

This section describes when WebAdmin checks for such anomalies, and what takes place when an anomaly is detected.

15.15.1 Port Number and Backup Storage Path Anomalies

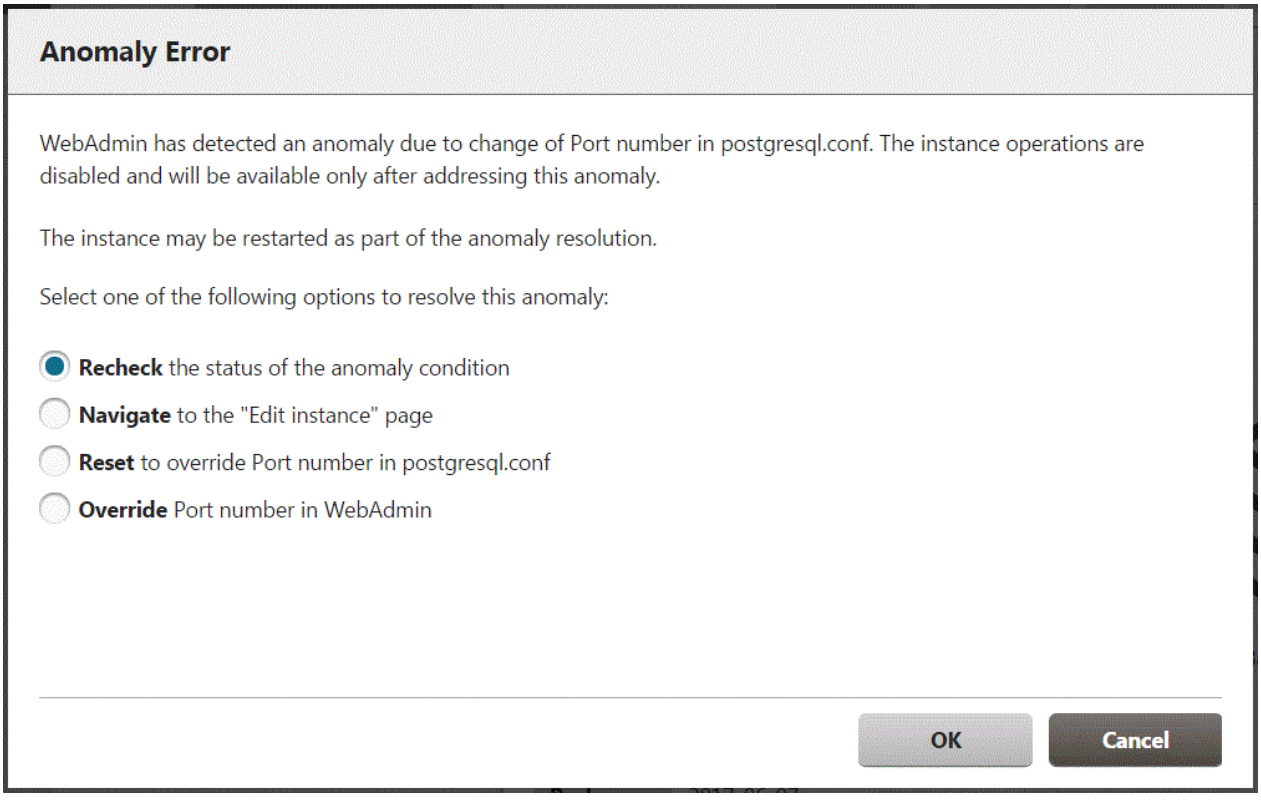
An anomaly occurs when the value of [Port number] and/or [Backup storage path] in WebAdmin is different from the value of its corresponding parameter in postgresql.conf - port and backup_destination, respectively.

WebAdmin checks for anomalies when an instance is selected for viewing or any instance operation is performed. Anomalies will be identified for the selected instance only.

The following occurs when an anomaly is detected in port number and/or backup storage path:

- All instance operation buttons are disabled, except for "Edit instance", "Refresh instance", and "Delete Mirroring Controller"
- A red error status indicator is displayed on the instance icon
- For an anomaly specific to backup storage path, a red error status indicator is displayed on the [Backup storage] disk icon, and [Backup storage status] is set to "Error"
- The message, "WebAdmin has detected an anomaly with...", is displayed in the [Message] section along with an associated [Solution] button

Click [Solution]. The [Anomaly Error] dialog box is displayed.



Select the required option, click [OK], and then resolve the anomaly error.

Refer to "Editing instance information" in the Installation and Setup Guide for Server for information on the [Edit instance] page.

Note

Critical errors encountered during anomaly resolution will be displayed, however, rollback of the instance to its previous state is not supported.

15.15.2 Mirroring Controller Anomalies

The following conditions will cause a Mirroring Controller anomaly:

- The Mirroring Controller management folder or configuration files have been deleted
- The permissions to the Mirroring Controller management folder or configuration files have been changed such that:
  - The instance administrator's access to Mirroring Controller configuration is denied
  - Users other than an instance administrator have access privileges to Mirroring Controller configuration files

WebAdmin checks for anomalies when Mirroring Controller status check is performed.

The following occurs when a Mirroring Controller anomaly is detected:

- All Mirroring Controller functionality is disabled for the replication cluster, except for "Delete Mirroring Controller"
- [Mirroring Controller status] is set to "Error"
- Either of the following messages is displayed in the [Message] section

  "Failed to access the Mirroring Controller management folder or configuration files '*path*'. Mirroring Controller functionality has been disabled. Consider deleting Mirroring Controller and adding it again."

  "Failed to find the Mirroring Controller management folder or configuration files '*path*'. Mirroring Controller functionality has been disabled. Consider deleting Mirroring Controller and adding it again."

Appendix A Parameters

This appendix describes the parameters to be set in the postgresql.conf file of Fujitsu Enterprise Postgres.

The postgresql.conf file is located in the data storage destination.

Information

The maximum value that can be expressed as a 4-byte signed integer changes according to the operating system. Follow the definition of the operating system in use.

Note

Note the following when specifying the path:

- Specify \\ as the path delimiter.
- Enclose the path in double quotes (") if it contains spaces.

- core_directory (string)

  This parameter specifies the directory where the corefile is to be output. If this parameter is omitted, the data storage destination is used by default. This parameter can only be set when specified on starting an instance. It cannot be changed dynamically, while an instance is active.

- core_contents (string)

  This parameter specifies the contents to be included in the corefile.

  - full: Outputs all contents of the server process memory to the corefile.
  - none: Does not output a corefile.
  - minimum: Outputs only non-shared memory server processes to the corefile. This reduces the size of the corefile. However, in some cases, this file may not contain sufficient information for examining the factor that caused the corefile to be output.

  If this parameter is omitted, "minimum" is used by default. This parameter can only be set when specified on starting an instance. It cannot be changed dynamically, while an instance is active.

- keystore_location (string)

  This parameter specifies the directory that stores the keystore file. Specify a different location from other database clusters. This parameter can only be set when specified on starting an instance. It cannot be changed dynamically, while an instance is active.

- tablespace_encryption_algorithm (string)

  This parameter specifies the encryption algorithm for tablespaces that will be created. Valid values are "AES128", "AES256", and "none". If you specify "none", encryption is not performed. The default value is "none". To perform encryption, it is recommended that you specify "AES256". Only superusers can change this setting.

- backup_destination (string)

  This parameter specifies the absolute path of the directory where pgx_dmpall will store the backup data. Specify a different location from other database clusters. This parameter can only be set when specified on starting an instance. It cannot be changed dynamically, while an instance is active.

  Place this directory on a different disk from the data directory to be backed up and the tablespace directory. Ensure that users do not store arbitrary files in this directory, because the contents of this directory are managed by the database system.

- search_path (string)

  When using the SUBSTR function compatible with Oracle databases, set "oracle" and "pg_catalog" in the search_path parameter. You must specify "oracle" before "pg_catalog".

Example

```
search_path = '"$user", public, oracle, pg_catalog'
```

Information

- The search_path feature specifies the priority of the schema search path. The SUBSTR function in Oracle database is defined in the oracle schema.
- Refer to "Statement Behavior" under "Server Administration" in the PostgreSQL Documentation for information on search_path.

- track_waits (string)

  This parameter enables collection of statistics for pgx_stat_lwlock and pgx_stat_latch.

  - on: Enables collection of statistics.
  - off: Disables collection of statistics.

  If this parameter is omitted, "on" is assumed.

  Only superusers can change this setting.

- track_sql (string)

  This parameter enables collection of statistics for pgx_stat_sql.

  - on: Enables collection of statistics.
  - off: Disables collection of statistics.

  If this parameter is omitted, "on" is assumed.

  Only superusers can change this setting.

Parameters for the in-memory feature

- reserve_buffer_ratio (numerical value)

  This parameter specifies the proportion of shared memory to be used for a stable buffer table.

  - Minimum value: 0
  - Maximum value: 80

  If this parameter is omitted, 0 will be used.

- vci.cost_threshold (numerical value)

  This parameter specifies the lowest cost that selects an execution plan that uses a VCI. If the cost of the best execution plan that does not use a VCI is lower than this value, that execution plan will be selected.

  - Minimum value: 0
  - Maximum value: Maximum value that can be expressed as a 4-byte signed integer

  If this parameter is omitted or a value outside this range is specified, 18000 will be used.

- vci.control_max_workers (numerical value)

  This parameter specifies the number of background workers that manage VCI. The number of workers for the entire instance is limited by max_worker_processes, so add the value specified here to max_worker_processes.

  - Minimum value: 1
  - Maximum value: 8388607

  If this parameter is omitted or a value outside this range is specified, 8 will be used.

- vci.enable (string)

  This parameter enables or disables VCI.

  - on: Enables VCI.
  - off: Disables VCI.

  If this parameter is omitted, "on" will be used.

- vci.log_query (string)

  This parameter enables or disables log output when VCI is not used due to insufficient memory specified by vci.max_local_ros.

  - on: Enables log output.
  - off: Disables log output.

  If this parameter is omitted, "off" will be used.

- vci.maintenance_work_mem (numerical value)

  This parameter specifies the maximum memory size used for maintenance of VCI (when executing CREATE INDEX, for example).

  - Minimum value: 1 MB
  - Maximum value: Maximum value that can be expressed as a 4-byte signed integer / 1024

  If this parameter is omitted or a value outside this range is specified, 256 MB will be used.

- vci.max_local_ros (numerical value)

  This parameter specifies the maximum memory size used for VCI scan.

  - Minimum value: 64 MB
  - Maximum value: Maximum value that can be expressed as a 4-byte signed integer

  If this parameter is omitted or a value outside this range is specified, 64 MB will be used.

- vci.max_parallel_degree (numerical value)

  This parameter specifies the maximum number of background workers used for parallel scan. The number of workers for the entire instance is limited by max_worker_processes, so add the value specified here to max_worker_processes.

  A value from -8388607 to 8388607 can be specified.

  - Integer (1 or greater): Parallel scan is performed using the specified degree of parallelism.
  - 0: Stops the parallel scan process.
  - Negative number: The specified value minus the maximum number of CPUs obtained from the environment is used as the degree of parallelism and parallel scan is performed.

  If this parameter is omitted or a value outside this range is specified, 0 will be used.

- vci.shared_work_mem (numerical value)

  This parameter specifies the maximum memory size used for VCI parallel scan.

  - Minimum value: 32 MB
  - Maximum value: Maximum value that can be expressed as a 4-byte signed integer

  If this parameter is omitted or a value outside this range is specified, 1 GB will be used.

- vci.smc_directory (string)

  This parameter specifies a directory name in which a temporary file is created as the dynamic shared memory during a scan using a VCI.

  If this parameter is omitted, a directory (*dataStorageDir*\\base\\pgsql_tmp) under the data storage directory will be used.

Parameters for the Global Meta Cache feature

- pgx_global_metacache (numerical value)

  Specifies the memory size of the GMC area.

  Specify a value calculated by the formula below.

  A value lower than the calculated value will still work, but the meta cache may not be able to fit into the GMC area.

  In this case, the system will discard the meta cache it thinks it is no longer needed, but if it is needed again, the meta cache will need to be expanded and will not perform well.

  If the value is less than 10 MB and is set to a nonzero value that disables the feature, the database startup fails because the Global Meta Cache feature cannot operate.

  A setting of 0 disables the Global Meta Cache feature. The default is 0.

  Changing this setting requires restarting the database.

```
Size of GMC area
 = Max(10MB,
     (All user table x 0.4 KB
      + All user Indexes x 0.3 KB
      + All user columns x 0.8 KB) x 1.5 (*1) )

*1) Safety Factor (1.5)
This value takes into account the case where both GMC before and after the change temporarily exist
at the same time in shared memory when the table definition is changed or the row of the system
catalog is changed.
```

- track_gmc (string)

  This parameter enables collection of statistics for pgx_stat_gmc.

  - on: Enables collection of statistics.
  - off: Disables collection of statistics.

  If this parameter is omitted, "on" is used.

  Only superusers can change this setting.

Parameters for the Local Meta Cache Limit feature

- pgx_catalog_cache_max_size(numerical value)

  Specifies the maximum amount of memory that the backend process should use as the catalog cache.

  You can enable catalog cache deletion by setting it to 8 KB or more.

  A setting of 0 disables the catalog cache removal. The default is 0.

  If no units are specified, they are treated as KB.

  - Minimum value: 8KB
  - Maximum value: Maximum value that can be expressed as a 4-byte signed integer

  When calculating the parameter settings, the factors that determine the cache size are calculated as the number of tables, the number of indexes, and the number of columns. What is kept as a catalog cache or relation cache also includes objects such as databases, roles, or procedures, but these are small compared to the above factors and do not need to be factored into them. It also includes a calculation method for pgx_relacion_cache_max_size because the given memory is distributed between the catalog cache and the relation cache.

Note

The calculation method here assumes that all backends have similar access and that the transaction also has access to a similar number of resources. If you have a small number of singular backends or transactions, consider excluding them as errors.

1. Determine how much memory a backend process can use. Decide by subtracting the memory size required by the entire system such as the database cache from the installed memory and dividing the rest by the number of connections.

2. For best performance, use the following formula to calculate the total memory size of the catalog cache when the backend holds the catalog cache for all resources accessed during its lifetime.
   The amount of memory varies depending on whether Global Meta Cache is enabled or disabled. Enabling Global Meta Cache reduces the amount of memory required because most of the cache is located on shared memory.

   ```
   When Global Meta Cache is enabled:
     (Number of tables to access + Number of indexes to access + Number of columns to access)
         × 0.1KB × 1.5 (*1)
   When Global Meta Cache is disabled:
     { Number of tables to access × 0.5KB(pg_class tuple size)
         + Number of indexes to access × 0.5KB(pg_index tuple size)
         + Number of columns to access × 1.0KB(pg_statistic tuple size)} × 1.5 (*1)

   *1) Safety Factor (1.5)
   The system catalog contains columns with variable-length types. For example, the tuple size in
   pg_class is a constant value multiplied by the number of tables, while relname in pg_class is
   variable length data.
   It is not practical to calculate every definition in detail, so we added 50% to the above
   formula.
   ```

3. In the same way as in 2., calculate the relation cache using the following formula.

   ```
     (1.4KB × Number of tables to access + 2.4KB × Number of indexes to access) × 1.5 (*1)

   *1) Safety Factor (1.5)
   The relation cache is structured to facilitate the use of table and index definitions, holds
   pointers to various objects, and is sized to include them. It is variable length because the
   type of object allocated by the table definition and its size change. Since it is not
   realistic to calculate for all definitions, 50% is added.
   ```

4. If the value of 1. ≥ the value of 2. + the value of 3., the backend process can keep all caches to the extent allowed, so there is no need to limit the caches. If you want to cap for safety, set the value of 2. to pgx_catalog_cache_max_size and the value of 3. to pgx_relation_cache_max_size.

5. If the value of 1. < the value of 2. + the value of 3. then you need to limit the cache. However, this parameter does not limit the size of the cache used by a transaction. Therefore, take the following steps.

6. Calculate the catalog cache used by a transaction using the formula in 2.

7. Calculate the relation cache used by a transaction using the formula in 3.

8. If the value of 1. < the value of 6. + the value of 7., then the value of 1. needs to be increased. In other words, in some cases, it may be necessary to increase the installed memory or reduce the number of connections.

9. If the value of 1. ≥ the value of 6. + the value of 7., the condition of 1. can be satisfied by limiting the cache with this parameter. Divide the value of 1. by the ratio of 2. and 3. and set it as a parameter. Set the value distributed to 2. to pgx_catalog_cache_max_size and the value distributed to 3. to pgx_relation_cache_max_size.

10. The value calculated in 9. is a provisional value. If you cannot meet your target performance, first try to shift the focus of allocation to the relation cache. This is because when executing SQL, the relation cache generated based on the catalog cache is mainly referenced, so it is advantageous to leave a large amount of relation cache. If the performance is still not satisfied, adjust the parameters by referring to "13.1.4 Performance Impact and Parameter Tuning of the Local Meta Cache Limit Feature".

Note

Be careful when partitioning the table.

The cached definition changes depending on whether the parent table is specified in the SQL statement or the child table is specified. In particular, note that if you specify a parent table, the definitions of all child tables are cached. This is because when you specify a parent table in an SQL statement, you need to know the definitions of all the child tables in order to determine which child table will contain the desired data. Note that the column information of the parent table is not cached.

When specifying the parent table:

```
Number of tables to access = Number of parent tables to access + Number of defined child tables
Number of columns = Number of defined columns x number of defined child tables
```

When specifying the child table directly:

```
Number of tables to access = Number of child tables actually accessed
Number of columns = Number of defined columns x number of child tables actually accessed
```

Example)

Suppose the parent table T (1 index, 3 columns) is split from child tables T1 to T5 (1 index, 3 columns, respectively). If the parent table T is specified in SQL, when the child tables that contain the data to be queried are limited to T1 and T2, and when accessing the data using the indexes defined by T1 and T2, calculate as follows.

```
Number of tables = 1(parent table) + 5(child table) = 6
Number of indexes = 2 (index to access)
Number of columns = 3 (number of columns) x 5 (child table) = 15
```

If you specify child tables T1 and T2 in SQL and use the indexes defined on T1 and T2 when accessing data, the calculation is as follows.

```
Number of tables = 2(child table)
Number of indexes = 2 (index to access)
Number of columns = 3 (number of columns) x 2 (child table) = 6
```

- pgx_relation_cache_max_size(numerical value)

  Specifies the maximum amount of memory that the backend process should use as the relation cache.

  You can enable catalog cache deletion by setting it to 8 KB or more.

  A setting of 0 disables the relation cache removal. The default is 0.

  If no units are specified, they are treated as KB.

  - Minimum value: 8KB
  - Maximum value: Maximum value that can be expressed as a 4-byte signed integer

  For the calculation method for parameter setting, refer to the calculation method of pgx_catalog_cache_max_size.

- pgx_cache_hit_log_interval(numerical value)

  Specifies the time interval to output a message indicating the cache reference status for each backend process.

  When the transaction ends, if the time set in this parameter has elapsed since the previous message was output, the message is output.

  If set to 0, a message will be output each time the transaction ends.

  Setting -1 disables the output. The default value is 10min.

  If no units are specified, they are treated as ms.

  Even if pgx_catalog_cache_max_size and pgx_relation_cache_max_size are disabled, the message output of the corresponding cache will be invalid.

  Immediately after connecting to the server, a small transaction occurs before the request from the user application, such as for user authentication. Since it is meaningless to know the hit rate for these, a message will be output at the end of the transaction that started after the time set in this parameter has elapsed after connecting to the server.

  For the same reason, setting a small value such as 0 may result in a message being printed at the end of such a small transaction.

  You can check which transaction the message corresponds to from the information output at the beginning.

  This information depends on the setting of the parameter log_line_prefix.

  - Minimum value: 0
  - Maximum value: 2147483647ms

See

Refer to "Server Configuration" under "Server Administration" in the PostgreSQL Documentation for information on other postgresql.conf parameters.

Appendix B System Administration Functions

This appendix describes the system administration functions of Fujitsu Enterprise Postgres.

See

Refer to "System Administration Functions" under "The SQL Language" in the PostgreSQL Documentation for information on other system administration functions.

B.1 WAL Mirroring Control Functions

The following table lists the functions that can be used for backup and recovery based on WAL mirroring.

Table B.1 WAL mirroring control functions

Name	Return type	Description
pgx_pause_wal_multiplexing()	void	Stops WAL multiplexing
pgx_resume_wal_multiplexing()	void	Resumes WAL multiplexing
pgx_is_wal_multiplexing_paused()	boolean	Returns true if WAL multiplexing has stopped

If WAL multiplexing has not been configured, these functions return an error. Setting the backup_destination parameter in postgresql.conf configures WAL multiplexing.

Only superusers can execute these functions.

B.2 Transparent Data Encryption Control Functions

The following table lists the functions that can be used for transparent data encryption.

Table B.2 Transparent data encryption control functions

Name	Return type	Description
pgx_open_keystore(*passphrase*)	void	Opens the keystore
pgx_set_master_key(*passphrase*)	void	Sets the master encryption key
pgx_set_keystore_passphrase(*oldPassphrase*, *newPassphrase*)	void	Changes the keystore passphrase

B.2.1 pgx_open_keystore

The pgx_open_keystore function uses the specified passphrase to open the keystore. When the keystore is opened, the master encryption key is loaded into the database server memory. In this way, you can access the encrypted data and create encrypted tablespaces. If the keystore is already open, this function returns an error.

Only superusers can execute this function. Also, this function cannot be executed within a transaction block.

B.2.2 pgx_set_master_key

The pgx_set_master_key function generates a master encryption key and stores it in keystore.

If the keystore does not exist, this function creates a keystore. If the keystore already exists, this function modifies the master encryption key. If the keystore has not been opened, this function opens it.

The passphrase is a string of 8 to 200 bytes.

Only superusers can execute this function. Also, this function cannot be executed within a transaction block. Processing is not affected by whether the keystore is open.

B.2.3 pgx_set_keystore_passphrase

The pgx_set_keystore_passphrase function changes the keystore passphrase.

Specify the current passphrase in *oldPassphrase*, and a new passphrase in *newPassphrase*.

The passphrase is a string of 8 to 200 bytes.

Only superusers can execute this function. Also, this function cannot be executed within a transaction block. Processing is not affected by whether the keystore is open.

B.3 Data Masking Control Functions

The table below lists the functions that can be used for data masking.

Table B.3 Data masking control functions

Name	Return type	Description
pgx_alter_confidential_policy	boolean	Changes masking policies
pgx_create_confidential_policy	boolean	Creates masking policies
pgx_drop_confidential_policy	boolean	Deletes masking policies
pgx_enable_confidential_policy	boolean	Enables or disables masking policies
pgx_update_confidential_values	boolean	Changes replacement characters when full masking is specified for masking type

B.3.1 pgx_alter_confidential_policy

Description

Changes masking policies

Format

The format varies depending on the content to be changed. The format is shown below.

- Common format

```
common_arg:
[schema_name    := 'schemaName',]
table_name    := 'tableName',
policy_name     := 'policyName'
```

- Add a masking target to a masking policy

```
pgx_alter_confidential_policy(
commonArg,
[action        := 'ADD_COLUMN', ]
column_name    := 'colName'
[, function_type    := 'FULL'] |
[, function_type    := 'PARTIAL', partialOpt] |
[, function_type    := 'REGEXP', regexpOpt]
)
```

```
partialOpt:
function_parameters    := 'maskingFmt'


regexpOpt:
regexp_pattern        := 'regexpPattern',
regexp_replacement    := 'regexpReplacementChar',
[, regexp_flags        := 'regexpFlags']
```

- Delete a masking target from a masking policy

```
pgx_alter_confidential_policy(
commonArg,
action        := 'DROP_COLUMN',
column_name    := 'colName'
)
```

- Change the masking condition

```
pgx_alter_confidential_policy(
commonArg,
action        := 'MODIFY_EXPRESSION',
expression    := 'expr'
)
```

- Change the content of a masking policy set for a masking target

```
pgx_alter_confidential_policy(
commonArg,
action        := 'MODIFY_COLUMN',
column_name    := 'colName'
[, function_type    := 'FULL'] |
[, function_type    := 'PARTIAL', partialOpt] |
[, function_type    := 'REGEXP', regexpOpt]
)


partialOpt:
function_parameters    := 'maskingFmt'


regexpOpt:
regexp_pattern        := 'regexpPattern',
regexp_replacement        := 'regexpReplacementChar',
[, regexp_flags        := 'regexpFlags']
```

- Change the masking policy description

```
pgx_alter_confidential_policy(
commonArg,
action        := 'SET_POLICY_DESCRIPTION',
policy_description    := 'policyDesc'
)
```

- Change the masking target description

```
pgx_alter_confidential_policy(
commonArg,
action        := 'SET_COLUMN_DESCRIPTION',
column_name    := 'colName',
column_description    := 'colDesc'
)
```

Argument

The argument varies depending on the content to be changed. Details are as follows.

- Common arguments

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	schema_name	varchar(63)	Schema name of table for which a masking policy is applied	'public'
	table_name	varchar(63)	Name of table for which a masking policy is applied	Mandatory
	policy_name	varchar(63)	Masking policy name	Mandatory

- Add a masking target to a masking policy

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	action	varchar(63)	'ADD_COLUMN'	'ADD_COLUMN'
	column_name	varchar(63)	Masking target name	Mandatory
	function_type	varchar(63)	Masking type - 'FULL': Full masking - 'PARTIAL': Partial masking - 'REGEXP': Regular expression masking	'FULL'
Partial masking	function_parameters	varchar(1024)	Masking format for partial masking	Mandatory
Regular expression masking	regexp_pattern	varchar(1024)	Search pattern for regular expression masking	Mandatory
	regexp_replacement	varchar(1024)	Replacement character/string for regular expression masking	Mandatory
	regexp_flags	varchar(20)	Regular expression flags	NULL

- Delete a masking target from a masking policy

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	action	varchar(63)	'DROP_COLUMN'	Mandatory
	column_name	varchar(63)	Masking target name	Mandatory

- Change the masking condition

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	action	varchar(63)	'MODIFY_EXPRESSION'	Mandatory

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
	expression	varchar(1024)	Masking condition to be changed	Mandatory

- Change the content of a masking policy set for a masking target

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	action	varchar(63)	'MODIFY_COLUMN'	Mandatory
	column_name	varchar(63)	Masking target name	Mandatory
	function_type	varchar(63)	Masking type - 'FULL': Full masking - 'PARTIAL': Partial masking - 'REGEXP': Regular expression masking	'FULL'
Partial masking	function_parameters	varchar(1024)	Masking format for partial masking	Mandatory
Regular expression masking	regexp_pattern	varchar(1024)	Search pattern for regular expression masking	Mandatory
	regexp_replacement	varchar(1024)	Replacement character/string for regular expression masking	Mandatory
	regexp_flags	varchar(20)	Regular expression flags	NULL

- Change the masking policy description

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	action	varchar(63)	'SET_POLICY_DESCRIPTION'	Mandatory
	policy_description	varchar(1024)	Masking policy description	Mandatory

- Change the masking target description

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	action	varchar(63)	'SET_COLUMN_DESCRIPTION'	Mandatory
	column_name	varchar(63)	Masking target name	Mandatory
	column_description	varchar(1024)	Masking target description	Mandatory

Details about whether arguments can be omitted are as follows.

Argument	Mandatory or optional									
	ADD_COLUMN			DROP_COLUMN	MODIFY_EXPRESSION	MODIFY_COLUMN			SET_POLICY_DESCRIPTION	SET_COLUMN_DESCRIPTION
	Full masking	Partial masking	Regular expression masking			Full masking	Partial masking	Regular expression masking		
schema_name	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
table_name	N	N	N	N	N	N	N	N	N	N
policy_name	N	N	N	N	N	N	N	N	N	N
action	Y	Y	Y	N	N	N	N	N	N	N
column_name	N	N	N	N	-	N	N	N	-	N
function_type	Y	N	N	-	-	Y	N	N	-	-
expression	-	-	-	-	N	-	-	-	-	-
policy_description	-	-	-	-	-	-	-	-	N	-
column_description	-	-	-	-	-	-	-	-	-	N
function_parameters	-	N	-	-	-	-	N	-	-	-
regexp_pattern	-	-	N	-	-	-	-	N	-	-
regexp_replacement	-	-	N	-	-	-	-	N	-	-
regexp_flags	-	-	Y	-	-	-	-	Y	-	-

Y: Can be omitted; N: Cannot be omitted; -: Ignored when specified

Return value

Return value	Description
TRUE	Ended normally
FALSE	Ended abnormally

Execution example 1

Adding masking policy p1 to masking target c2

```
postgres=# select pgx_alter_confidential_policy(table_name := 't1', policy_name := 'p1', action :=
'ADD_COLUMN', column_name := 'c2', function_type := 'PARTIAL', function_parameters := 'VVVFVVVVFVVVV,
VVV-VVVV-VVVV, *, 4, 11');
 pgx_alter_confidential_policy
-------------------------------
 t
(1 row)
```

Execution example 2

Deleting masking target c1 from masking policy p1

```
postgres=# select pgx_alter_confidential_policy(table_name := 't1', policy_name := 'p1', action :=
'DROP_COLUMN', column_name := 'c1');
 pgx_alter_confidential_policy
-------------------------------
```

```
 t
(1 row)
```

Execution example 3

Changing the masking condition for masking policy p1

```
postgres=# select pgx_alter_confidential_policy(table_name := 't1', policy_name := 'p1', action :=
'MODIFY_EXPRESSION', expression := 'false');
 pgx_alter_confidential_policy
-------------------------------
 t
(1 row)
```

Execution example 4

Changing the content of masking policy p1 set for masking target c2

```
postgres=# select pgx_alter_confidential_policy(table_name := 't1', policy_name := 'p1', action :=
'MODIFY_COLUMN', column_name := 'c2', function_type := 'FULL');
 pgx_alter_confidential_policy
-------------------------------
 t
(1 row)
```

Execution example 5

Changing the description of masking policy p1

```
postgres=# select pgx_alter_confidential_policy(table_name := 't1', policy_name := 'p1', action :=
'SET_POLICY_DESCRIPTION', policy_description := 'this policy is an example.');
 pgx_alter_confidential_policy
-------------------------------
 t
(1 row)
```

Execution example 6

Changing the description of masking target c2

```
postgres=# select pgx_alter_confidential_policy(table_name := 't1', policy_name := 'p1', action :=
'SET_COLUMN_DESCRIPTION', column_name := 'c2', column_description := 'c2 column is FULL.');
 pgx_alter_confidential_policy
-------------------------------
 t
(1 row)
```

Description

- The arguments for the pgx_alter_confidential_policy system management function can be specified in any order.
- The action parameters below can be specified. When action parameters are omitted, ADD_COLUMN is applied.

Parameter	Description
ADD_COLUMN	Adds a masking target to a masking policy.
DROP_COLUMN	Deletes a masking target from a masking policy.
MODIFY_EXPRESSION	Changes expression.
MODIFY_COLUMN	Changes the content of a masking policy set for a masking target.
SET_POLICY_DESCRIPTION	Changes policy_description.

Parameter	Description
SET_COLUMN_DESCRIPTION	Changes column_description.

- The function_parameters argument is enabled when the function_type is PARTIAL. If the function_type is other than PARTIAL, it will be ignored.
- The arguments below are enabled when the function_type is REGEXP. If the function_type is other than REGEXP, these arguments will be ignored.
  - regexp_pattern
  - regexp_replacement
  - regexp_flags

See

- Refer to "String Constants" in the PostgreSQL Documentation for information on the strings to specify for arguments.
- Refer to "POSIX Regular Expressions" in the PostgreSQL Documentation and check pattern, replacement, and flags for information on the values that can be specified for regexp_pattern, regexp_replacement, and regexp_flags.

B.3.2 pgx_create_confidential_policy

Description

Creates masking policies

Format

The format varies depending on the masking type. The format is shown below.

```
pgx_create_confidential_policy(
[schema_name        := 'schemaName',]
table_name        := 'tableName',
policy_name         := 'policyName',
expression        := 'expr'
[, enable            := 'policyStatus']
[, policy_description    := 'policyDesc']
[, column_name        := 'colName'
     [, function_type    := 'FULL'] |
     [, function_type    := 'PARTIAL', partialOpt] |
     [, function_type    := 'REGEXP', regexpOpt]
     [, column_description    := 'colDesc']
])


partialOpt:
function_parameters    := 'maskingFmt'


regexpOpt:
regexp_pattern        := 'regexpPattern',
regexp_replacement        := 'regexpReplacementChar',
[, regexp_flags        := 'regexpFlags']
```

Argument

Details are as follows.

Masking type for which an argument can be specified	Argument	Data type	Description	Default value
All	schema_name	varchar(63)	Schema name of table for which the masking policy is created	'public'
	table_name	varchar(63)	Name of table for which the masking policy is created	Mandatory
	policy_name	varchar(63)	Masking policy name	Mandatory
	expression	varchar(1024)	Masking condition	Mandatory
	enable	boolean	Masking policy status - 't': Enabled - 'f': Disabled	't'
	policy_description	varchar(1024)	Masking policy description	NULL
	column_name	varchar(63)	Masking target name	NULL
	function_type	varchar(63)	Masking type - 'FULL': Full masking - 'PARTIAL': Partial masking - 'REGEXP': Regular expression masking	'FULL'
	column_description	varchar(1024)	Masking target description	NULL
Partial masking	function_parameters	varchar(1024)	Masking format for partial masking	Mandatory
Regular expression masking	regexp_pattern	varchar(1024)	Search pattern for regular expression masking	Mandatory
	regexp_replacement	varchar(1024)	Replacement character/string for regular expression masking	Mandatory
	regexp_flags	varchar(20)	Regular expression flags	NULL

Details about whether arguments can be omitted are as follows.

Argument	Mandatory or optional		
	Full masking	Partial masking	Regular expression masking
schema_name	Y	Y	Y
table_name	N	N	N
policy_name	N	N	N
expression	N	N	N
enable	Y	Y	Y
policy_description	Y	Y	Y
column_name	Y	Y	Y
function_type	Y	Y	Y
column_description	Y	Y	Y
function_parameters	-	N	-
regexp_pattern	-	-	N
regexp_replacement	-	-	N

Argument	Mandatory or optional		
	Full masking	Partial masking	Regular expression masking
regexp_flags	-	-	Y

Y: Can be omitted; N: Cannot be omitted; -: Ignored when specified

Return value

Return value	Description
TRUE	Ended normally
FALSE	Ended abnormally

Execution example 1

Creating masking policy p1 that does not contain a masking target

```
postgres=# select pgx_create_confidential_policy(table_name := 't1', policy_name := 'p1',
expression := '1=1');
 pgx_create_confidential_policy
--------------------------------
 t
(1 row)
```

Execution example 2

Creating masking policy p1 that contains masking target c1 of which the masking type is full masking

```
postgres=# select pgx_create_confidential_policy(schema_name := 'public', table_name := 't1',
policy_name := 'p1', expression := '1=1', enable := 't', policy_description := 'this policy is an
example.', column_name := 'c1', function_type := 'FULL', column_description := 'c1 column is FULL.');
 pgx_create_confidential_policy
--------------------------------
 t
(1 row)
```

Execution example 3

Creating masking policy p1 that contains masking target c2 of which the masking type is partial masking

```
postgres=# select pgx_create_confidential_policy( table_name := 't1', policy_name := 'p1',
expression := '1=1', column_name := 'c2', function_type := 'PARTIAL', function_parameters :=
'VVVFVVVVFVVVV, VVV-VVVV-VVVV, *, 4, 11');
 pgx_create_confidential_policy
--------------------------------
 t
(1 row)
```

Execution example 4

Creating masking policy p1 that contains masking target c3 of which the masking type is regular expression masking

```
postgres=# select pgx_create_confidential_policy( table_name := 't1', policy_name := 'p1',
expression := '1=1', column_name := 'c3', function_type := 'REGEXP', regexp_pattern := '(.*)(@.*)',
regexp_replacement := 'xxx\2', regexp_flags := 'g');
 pgx_create_confidential_policy
--------------------------------
 t
(1 row)
```

Description

- The arguments for the pgx_create_confidential_policy system management function can be specified in any order.
- If column_name is omitted, only masking policies that do not contain masking target will be created.
- One masking policy can be created for each table. Use the pgx_alter_confidential_policy system management function to add a masking target to a masking policy.
- The function_parameters argument is enabled when the function_type is PARTIAL. If the function_type is other than PARTIAL, it will be ignored.
- The arguments below are enabled when the function_type is REGEXP. If the function_type is other than REGEXP, these arguments will be ignored.
  - regexp_pattern
  - regexp_replacement
  - regexp_flags

Note

If a table for which a masking policy is to be applied is deleted, delete the masking policy as well.

See

- Refer to "String Constants" in the PostgreSQL Documentation for information on the strings to specify for arguments.
- Refer to "POSIX Regular Expressions" in the PostgreSQL Documentation and check pattern, replacement, and flags for information on the values that can be specified for regexp_pattern, regexp_replacement, and regexp_flags.

B.3.3 pgx_drop_confidential_policy

Description

Deletes masking policies

Format

```
pgx_drop_confidential_policy(
[schema_name    := 'schemaName', ]
table_name     := 'tableName',
policy_name    := 'policyName'
)
```

Argument

Details are as follows.

Argument	Data type	Description	Default value
schema_name	varchar(63)	Schema name of table for which a masking policy is deleted	'public'
table_name	varchar(63)	Name of table for which a masking policy is deleted	Mandatory
policy_name	varchar(63)	Masking policy name	Mandatory

Details about whether arguments can be omitted are as follows.

Argument	Mandatory or optional
schema_name	Y
table_name	N
policy_name	N

Y: Can be omitted; N: Cannot be omitted

Return value

Return value	Description
TRUE	Ended normally
FALSE	Ended abnormally

Execution example

Deleting masking policy p1

```
postgres=# select pgx_drop_confidential_policy(table_name := 't1', policy_name := 'p1');
 pgx_drop_confidential_policy
------------------------------
 t
(1 row)
```

Description

The arguments for the pgx_drop_confidential_policy system management function can be specified in any order.

Note

If a table for which a masking policy is to be applied is deleted, delete the masking policy as well.

See

Refer to "String Constants" in the PostgreSQL Documentation for information on the strings to specify for arguments.

B.3.4 pgx_enable_confidential_policy

Description

Enables or disables masking policies

Format

```
pgx_enable_confidential_policy(
[schema_name    := 'schemaName', ]
table_name     := 'tableName',
policy_name    := 'policyName',
enable        := 'policyStatus'
)
```

Argument

Details are as follows.

Argument	Data type	Description	Default value
schema_name	varchar(63)	Schema name of table for which a masking policy is enabled or disabled	'public'
table_name	varchar(63)	Name of table for which a masking policy is enabled or disabled	Mandatory
policy_name	varchar(63)	Masking policy name	Mandatory
enable	boolean	Masking policy status - 't': Enabled - 'f': Disabled	Mandatory

Details about whether arguments can be omitted are as follows.

Argument	Mandatory or optional
schema_name	Y
table_name	N
policy_name	N
enable	N

Y: Can be omitted; N: Cannot be omitted

Return value

Return value	Description
TRUE	Ended normally
FALSE	Ended abnormally

Execution example

Enabling masking policy p1

```
postgres=# select pgx_enable_confidential_policy(table_name := 't1', policy_name := 'p1', enable :=
't');
 pgx_enable_confidential_policy
--------------------------------
 t
(1 row)
```

Description

The arguments for the pgx_enable_confidential_policy system management function can be specified in any order.

See

Refer to "String Constants" in the PostgreSQL Documentation for information on the strings to specify for arguments.

B.3.5 pgx_update_confidential_values

Description

Changes replacement characters when full masking is specified for masking type

Format

```
pgx_update_confidential_values(
[number_value    := 'numberValue']
[, char_value    := 'charValue']
[, varchar_value    := 'varcharValue']
[, date_value    := 'dateValue']
[, ts_value    := 'tsValue']
)
```

Argument

Details are as follows.

Argument	Data type	Description
number_value	integer	Replacement character in numeric type
char_value	varchar(1)	Replacement character in char type
varchar_value	varchar(1)	Replacement character in varchar type
date_value	date	Replacement character in date type
ts_value	timestamp	Replacement character in timestamp type

Return value

Return value	Description
TRUE	Ended normally
FALSE	Ended abnormally

Execution example

Using '*' as a replacement character in char type and varchar type

```
postgres=# select pgx_update_confidential_values(char_value := '*', varchar_value := '*');
 pgx_update_confidential_values
--------------------------------
 t
(1 row)
```

Description

- The arguments for the pgx_update_confidential_values system management function can be specified in any order.
- Specify one or more arguments for the pgx_update_confidential_values system management function. A replacement character is not changed for an omitted argument.

Refer to "String Constants" in the PostgreSQL Documentation for information on the strings to specify for arguments.

B.4 VCI Data Load Control Function

The table below lists the function that loads VCI data to buffer cache.

Table B.4 VCI data load control function

Name	Return type	Description
pgx_prewarm_vci(vci_index regclass)	int8	Loads the VCI data to buffer cache.

pgx_prewarm_vci loads the specified VCI data to buffer cache and returns the number of blocks of the loaded VCI data.

The aggregation process using VCI may take time immediately after an instance is started, because the VCI data has not been loaded to buffer cache. Therefore, the first aggregation process can be sped up by executing pgx_prewarm_vci after an instance is started.

The amount of memory required for preloading is the number of blocks returned by pgx_prewarm_vci multiplied by the size of one block.

This function can only be executed if the user has reference privilege to the VCI index and execution privilege to the pg_prewarm function.

B.5 High-Speed Data Load Control Functions

The table below lists the functions that can be used for high-speed data load.

Table B.5 High-speed data load control functions

Name	Return type	Description
pgx_loader	bigint	Creates dynamic shared memory, starts parallel workers and loads data
pgx_loader_recovery	smallint	Resolves in-doubt transactions

The pgx_loader command executes the above functions internally.

Appendix C System Views

This appendix describes how to use the system views in Fujitsu Enterprise Postgres.

See

Refer to "System Views" under "Internals" in the PostgreSQL Documentation for information on other system views.

C.1 pgx_tablespaces

The pgx_tablespaces view provides information related to the encryption of tablespaces.

Table C.1 pgx_tablespaces view

Column	Type	References	Description
spctablespace	oid	pg_tablespace.oid	Tablespace OID
spcencalgo	text		Tablespace encryption algorithm

The spcencalgo string displays one of the following values:

- none: Tablespace is not encrypted
- AES128: AES with key length of 128 bits
- AES256: AES with key length of 256 bits

C.2 pgx_stat_lwlock

The pgx_stat_lwlock view displays statistics related to lightweight locks, with each type of content displayed on a separate line.

Table C.2 pgx_stat_lwlock view

Column	Type	Description
lwlock_name	name	Name of the lightweight lock
total_waits	bigint	Number of waits caused by the lightweight lock
total_wait_time	double precision	Number of milliseconds spent in waits caused by the lightweight lock
stats_reset	timestamp with timezone	Last time at which this statistics was reset

C.3 pgx_stat_latch

The pgx_stat_latch view displays statistics related to latches, with each type of wait information within Fujitsu Enterprise Postgres displayed on a separate line.

Table C.3 pgx_stat_latch view

Column	Type	Description
latch_name	name	Name of the latch
total_waits	bigint	Number of waits caused a wait
total_wait_time	double precision	Number of milliseconds spent in waits caused by the latch
stats_reset	timestamp with timezone	Last time at which this statistic was reset

C.4 pgx_stat_walwriter

The pgx_stat_walwriter view displays statistics related to WAL writing, in a single line.

Table C.4 pgx_stat_walwriter view

Column	Type	Description
dirty_writes	bigint	Number of times old WAL buffers were written to the disk because the WAL buffer was full when WAL records were added
writes	bigint	Number of WAL writes
write_blocks	bigint	Number of WAL write blocks
total_write_time	double precision	Number of milliseconds spent on WAL writing
stats_reset	timestamp with timezone	Last time at which this statistic was reset

C.5 pgx_stat_sql

The pgx_stat_sql view displays statistics related to SQL statement executions, with each type of SQL statement displayed on a separate line.

Table C.5 pgx_stat_sql view

Column	Type	Description
selects	bigint	Number of SELECT statements executed In database multiplexing mode, this number includes the SELECT statements executed in Mirroring Controller. Mirroring Controller executes the SELECT statement using the interval specified for the heartbeat_interval of the server definition file (milliseconds).
inserts	bigint	Number of INSERT statements executed
deletes	bigint	Number of DELETE statements executed
updates	bigint	Number of UPDATE statements executed
selects_with_parallelism	bigint	Number of times parallel scan was used in SELECT statements
inserts_with_parallelism	bigint	Not used
deletes_with_parallelism	bigint	Not used
updates_with_parallelism	bigint	Not used
copies_with_parallelism	bigint	Not used
declares	bigint	Number of DECLARE statements executed (number of cursor OPENs)
fetches	bigint	Number of FETCH statements executed
checkpoints	bigint	Number of CHECKPOINT statements executed
clusters	bigint	Number of CLUSTER statements executed
copies	bigint	Number of COPY statements executed
reindexes	bigint	Number of REINDEX statements executed
truncates	bigint	Number of TRUNCATE statements executed
locks	bigint	Number of times a lock occurred
stats_reset	timestamp with timezone	Last time at which this statistic was reset

C.6 pgx_stat_gmc

The pgx_stat_gmc view provides information about the GMC areas.

Table C.6 pgx_stat_gmc view

Column	Type	Description
searches	bigint	Number of times the cache table is searched.
hits	bigint	Number of times the cache table is hit.
size	bigint	The current amount of memory (bytes) used in the GMC area.
stats_reset	timestamp with timezone	Last time these statistics were reset.

C.7 pgx_stat_progress_loader

The pgx_stat_progress_loader view provides overall progress information for pgx_loader command.

The pgx_stat_progress_loader view displays the sum of the progress information of the back-end processes and the number of parallel worker processes when pgx_loader runs.

Table C.7 pgx_stat_progress_loader view

Column	Type	Description
pid	integer	Process ID of the backend.
datid	Oid	Oid of the database to connect to.
datname	name	Name of the database to connect to.
relid	Oid	Oid of the table to load.
command	text	Command executes the load process. (Always "COPY FROM"for pgx_loader)
type	text	Type of data source for the load operation.
bytes_processed	bigint	Size of the data at the end of the load. (Backend and worker totals)
bytes_total	bigint	Size of the data to load. (Backend and worker totals)
tuples_processed	bigint	Number of rows that have completed loading. (Backend and worker totals)
tuples_excluded	bigint	Number of rows skipped during the load process. (Backend and worker totals)

Appendix D Tables Used by Data Masking

This appendix explains tables used by the data masking feature.

Note

These tables are updated by the data masking control function, so do not use SQL statements to directly update these tables.

D.1 pgx_confidential_columns

This table provides information on masking target for which masking policies are set.

Column	Type	Description
schema_name	varchar(63)	Schema name of table for which a masking policy is applied
table_name	varchar(63)	Name of table for which a masking policy is applied
policy_name	varchar(63)	Masking policy name
column_name	varchar(63)	Masking target name
function_type	varchar(63)	Masking type - 'FULL': Full masking - 'PARTIAL': Partial masking - 'REGEXP': Regular expression masking
function_parameters	varchar(1024)	Masking format for partial masking
regexp_pattern	varchar(1024)	Search pattern for regular expression masking
regexp_replacement	varchar(1024)	Replacement character/string for regular expression masking
regexp_flags	varchar(20)	Regular expression flags
column_description	varchar(1024)	Masking target description

Execution example

```
postgres=# select * from pgx_confidential_columns;
 schema_name | table_name | policy_name | column_name | function_type |
function_parameters           | regexp_pattern | regexp_replacement | regexp_flags |
column_description
-------------+------------+-------------+-------------+---------------
+---------------------------------------+----------------+--------------------+-------------
+--------------------
 public      | t1         | p1          | c1          | FULL          |
|                |                     |             |
 public      | t1         | p1          | c2          | PARTIAL       | VVVFVVVVFVVVV, VVV-VVVV-VVVV,
*, 4, 11 |                |                     |             |
(2  row)
```

D.2 pgx_confidential_policies

This table provides information on masking policies.

Column	Type	Description
schema_name	varchar(63)	Schema name of table for which a masking policy is applied

Column	Type	Description
table_name	varchar(63)	Name of table for which a masking policy is applied
policy_name	varchar(63)	Masking policy name
expression	varchar(1024)	Masking condition
enable	boolean	Masking policy status - 't': Enabled - 'f': Disabled
policy_description	varchar(1024)	Masking policy description

Execution example

```
postgres=# select * from pgx_confidential_policies;
 schema_name | table_name | policy_name | expression | enable | policy_description
-------------+------------+-------------+------------+--------+--------------------
 public      | t1         | p1          | 1=1        | t      |
(1 row)
```

D.3 pgx_confidential_values

This table provides information on replacement characters when full masking is specified for masking type.

Column	Data type	Description	Default value
number_value	integer	Numeric	0
char_value	varchar(1)	char type	Spaces
varchar_value	varchar(1)	varchar type	Spaces
date_value	date	date type	'1970-01-01'
timestamp_value	timestamp	timestamp type	'1970-01-01 00:00:00'

Execution example

```
postgres=# select * from pgx_confidential_values;
 number_value | char_value | varchar_value | date_value |      ts_value
--------------+------------+---------------+------------+---------------------
            0 |            |               | 1970-01-01 | 1970-01-01 00:00:00
(1 row)
```

Appendix E Tables Used by High-Speed Data Load

This appendix describes the tables used by high-speed data load.

E.1 pgx_loader_state

The pgx_loader_state table provides information about transactions prepared by high-speed data load.

Column	Type	Description
id	serial	Unique identifier. This value is assigned from the pgx_loader_state_id_seq sequence.
gid	text	Global transaction identifier assigned to a transaction.
state	text	State of the transaction. The value can be one of the following: - commit: The prepared transaction has been committed. - rollback: The prepared transaction is in in-doubt state.
leader_pid	integer	Process ID of the backend process (leader process) that executed the pgx_loader control function.
role_oid	integer	Role identifier (OID). A prepared transaction can only be completed by the same user who executed the original transaction or by a superuser.
relation_oid	integer	Object identifier (OID).

Note

The pgx_loader_state table and pgx_loader_state_id_seq sequence are updated by high-speed data load. Do not update these database objects directly using SQL.

Appendix F Starting and Stopping the Web Server Feature of WebAdmin

To use WebAdmin for creating and managing a Fujitsu Enterprise Postgres instance on a server where Fujitsu Enterprise Postgres is installed, you must first start the Web server feature of WebAdmin.

- Using WebAdmin in a single-server configuration

  You must start the Web server on the server on which Fujitsu Enterprise Postgres and WebAdmin are installed.

- Using WebAdmin in a multiserver configuration

  You must start the Web server on all servers on which WebAdmin has been installed.

This appendix describes how to start and stop the Web server feature of WebAdmin.

See

Refer to "Installing WebAdmin in a Multiserver Configuration" in the Installation and Setup Guide for Server for information on multiserver installation.

F.1 Starting the Web Server Feature of WebAdmin

Follow the procedure below to start the Web server feature of WebAdmin:

1. Display the [Services] window

   In the [Administrative Tools], click [Services].

2. Start a service

   Select the displayed name "Fujitsu Enterprise Postgres WebAdmin *version*", and then click [Start Service].

You can also start a service by specifying the service name of the Web server feature of WebAdmin in the net start command or sc start command.

F.2 Stopping the Web Server Feature of WebAdmin

This section describes how to stop the Web server feature of WebAdmin.

Follow the procedure below to stop the Web server feature of WebAdmin:

1. Display the [Services] window

   In the [Administrative Tools], click [Services].

2. Stop a service

   Select the displayed name "Fujitsu Enterprise Postgres WebAdmin *version*", and then click [Stop Service].

You can also stop a service by specifying the service name of the Web server feature of WebAdmin in the net stop command or sc stop command.

Appendix G WebAdmin Wallet

This appendix describes how to use the Wallet feature of WebAdmin.

When a remote instance or a standby instance is created, it is necessary to provide user name and password for authentication with the remote machine or the database instance.

The Wallet feature in WebAdmin is a convenient way to create and store these credentials.

Once created, these credentials can be repeatedly used in one or more instances.

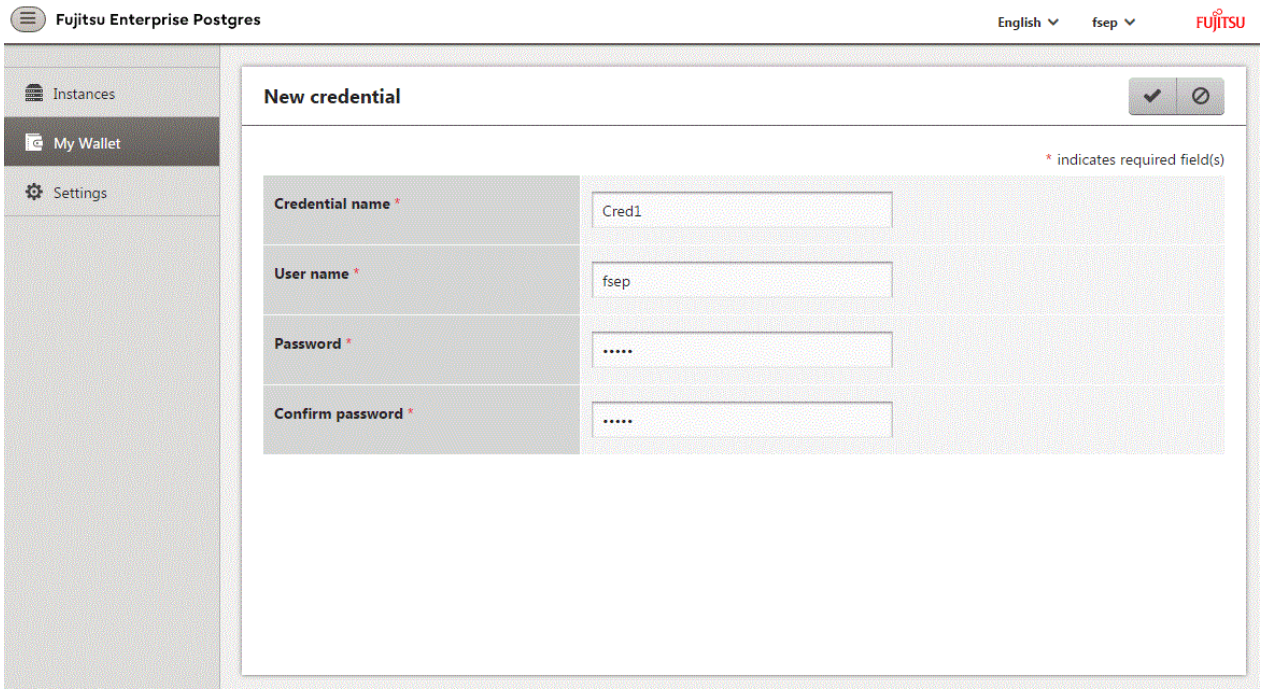
Note

It is not mandatory to create a credential in the Wallet. It is possible to create a remote instance or a standby instance without creating any credential in the Wallet.

If no credential is created beforehand, a user name and password can be entered in the instance creation page. When creating a "Remote" instance, if operating system credentials are entered without using a credential stored in the Wallet, WebAdmin automatically creates a credential with the given user name and password, and stores it in the user's wallet for future use.

G.1 Creating a Credential

1. In the [My Wallet] tab, click ➕. The [New credential] page will be displayed.
2. Enter the information for the credentials.



Enter the following items. Credential name, User name and Password should not contain hazardous characters. Refer to "Appendix H WebAdmin Disallow User Inputs Containing Hazardous Characters".

- [Credential name]: Name of the credential

  The name must meet the conditions below:

  - Maximum of 16 characters
  - The first character must be an ASCII alphabetic character
  - The other characters must be ASCII alphanumeric characters

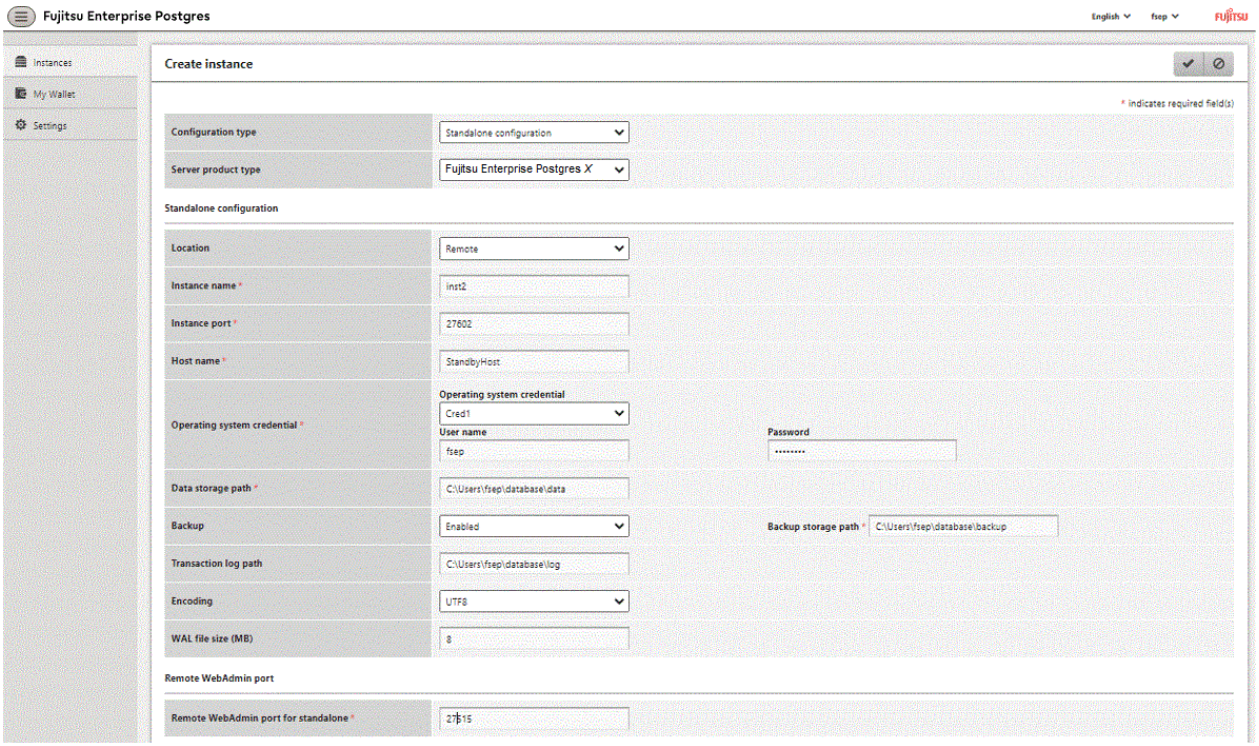
- [User name]: The operating system user name or database instance user name that will be used later
- [Password]: Password for the user
- [Confirm password]: Reenter the password.

3. Click ✔ to store the credential.

G.2 Using a Credential

Once a credential is created in the Wallet, it can be used during remote instance creation or standby instance creation.

The following page uses the credential that was created in the previous section.



When "Cred1" is selected in [Operating system credential], the user name and password are automatically populated from the credential.

Appendix H WebAdmin Disallow User Inputs Containing Hazardous Characters

WebAdmin considers the following as hazardous characters, which are not allowed in user inputs.

```
| (pipe sign)

& (ampersand sign)

; (semicolon sign)

$ (dollar sign)

% (percent sign)

@ (at sign)

' (single apostrophe)

" (quotation mark)

\' (backslash-escaped apostrophe)

\" (backslash-escaped quotation mark)

<> (triangular parenthesis)

() (parenthesis)

+ (plus sign)

CR (Carriage return, ASCII 0x0d)

LF (Line feed, ASCII 0x0a)

, (comma sign)

\ (backslash)
```

Appendix I Collecting Failure Investigation Data

If the cause of an error that occurs while building the environment or during operations is unclear, data must be collected for initial investigation.

This appendix describes how to collect data for initial investigation.

Use FJQSS (Information Collection Tool) to collect data for initial investigation.

See

Refer to the following manual for information on how to use FJQSS.

- In the [Apps] menu, select [FJQSS (Information Collection Tool)], and then click [FJQSS User's Guide].

Note

- When using FJQSS to collect data for initial investigation, a window will be displayed for you to set the following environment variables:

  - PGDATA

    Set the data storage destination.

  - PGDATABASE

    Set the database name from which you want to collect data for initial investigation.

  - PGPORT

    Set the instance port number. This does not need to be set if the default port number (27500) has not been changed.

  - PGUSER

    Set the database superuser.
    Set the database superuser so that client authentication is possible.
    FJQSS establishes a TCP/IP connection with the template1 database and collects data from the database.

- Refer to "Collecting Failure Investigation Data" in the Cluster Operation Guide (Database Multiplexing) for information on how to collect failure investigation data when performing database multiplexing.

Information

FJQSS corresponds the required operating system of Fujitsu Enterprise Postgres described below.

- "Required Operating System" in the Installation and Setup Guide for Server.

Index

[A]

Actions in Response to Instance Startup Failure....121
All user data within the specified tablespace....27
Approximate backup time....18
Approximate recovery time....95
Automatically opening the keystore....38

[B]

Backing Up and Recovering the Keystore....32
Backing Up and Restoring/Recovering the Database....34
Backup/Recovery Using the Copy Command....86
Backup and recovery using the pgx_dmpall and pgx_rcvall commands....34
backup cycle....19
Backup data....27
Backup operation....20
Backup operation (file backup)....21
Backup status....20,22
Backup using the backup information file....87
Backup Using the Copy Command....89
backup_destination (string)....128
Building and starting a standby server....38

[C]

Changing a Masking Policy....46
Changing the Keystore Passphrase....31
Changing the Master Encryption Key....31
Changing the master encryption key and the passphrase....38
Checking an Encrypted Tablespace....30
Checking backup status....90
Checking the operating state of an instance....16
Checking the operating status of an instance....14
Collecting Failure Investigation Data....160
Configuration of the Copy Command....86
Configuration of the copy command for backup....88
Configuration of the copy command for recovery....88
Confirming a Masking Policy....46
Continuous archiving and point-in-time recovery....35
Copy Command for Backup....91
Copy Command for Recovery....93
Copy Command Interface....91
core_contents (string)....128
core_directory (string)....128
Creating a Masking Policy....45
Cyclic usage of the backup area....86

[D]

Data Masking....40
Data Types for Masking....48
Deleting a Masking Policy....48
Determining the backup area of the latest backup....90

[E]

Enabling and Disabling a Masking Policy....47
Enabling Automatic Opening of the Keystore....31
Encrypting a Tablespace....29
Encrypting Existing Data....36
Encryption mechanisms....27
Errors in More Than One Storage Disk....120

[F]

Faster encryption and decryption based on hardware....27
File system level backup and restore....35

[H]

High-Speed Data Load....74

[I]

If failure occurred in the data storage disk or the transaction log storage disk....96
If failure occurred on the backup data storage disk....97,99
If failure occurred on the data storage disk or the transaction log storage directory....97
Importing and Exporting the Database....36
Installing and Operating the In-memory Feature....63

[K]

keystore_location (string)....128

[L]

Logging in to WebAdmin....3
log in....3

[M]

Managing the Keystore....31
Masking Condition....41
Masking Format....42
Masking Policy....40
Masking Target....41
Masking Type....41
Monitoring Database Activity....54

[O]

Opening the Keystore....28
Operating Fujitsu Enterprise Postgres....1

[P]

Parallel Query....72
Performing backup....89
Perform recovery....90
Periodic Backup....19
pgx_global_metacache (numerical value)....131
pgx_stat_gmc view....152
pgx_stat_latch view....150
pgx_stat_lwlock view....150
pgx_stat_progress_loader view....152
pgx_stat_sql view....151
pgx_stat_walwriter view....151
pgx_tablespaces....150
pgx_tablespaces view....150
Placement and automatic opening of the keystore file....37
Placing the keystore file....37
Preparing for backup....89

[R]
Recovery Using the Copy Command....90
reserve_buffer_ratio (numerical value)....129

[S]
Scope of encryption....27
search_path (string)....128
Security-Related Notes....38
Security Notes....49
Setting a restore point....22
Setting the Master Encryption Key....28
Starting and Stopping the Web Server Feature of WebAdmin....156
Starting an instance....13,15
Starting pgAdmin....3,4
Startup URL for WebAdmin....3
Stopping an instance....13,15
Streaming replication support....28
Streaming Replication Using WebAdmin....58
Strong encryption algorithms....27
System Administration Functions....135
System Views....150

[T]
tablespace_encryption_algorithm (string)....128
Tables Used by Data Masking....153
Tips for Installing Built Applications....39
track_gmc (string)....131
track_sql (string)....129
track_waits (string)....129
Transparent Data Encryption Control Functions....135
Two-layer encryption key and the keystore....27

[U]
User environment....3

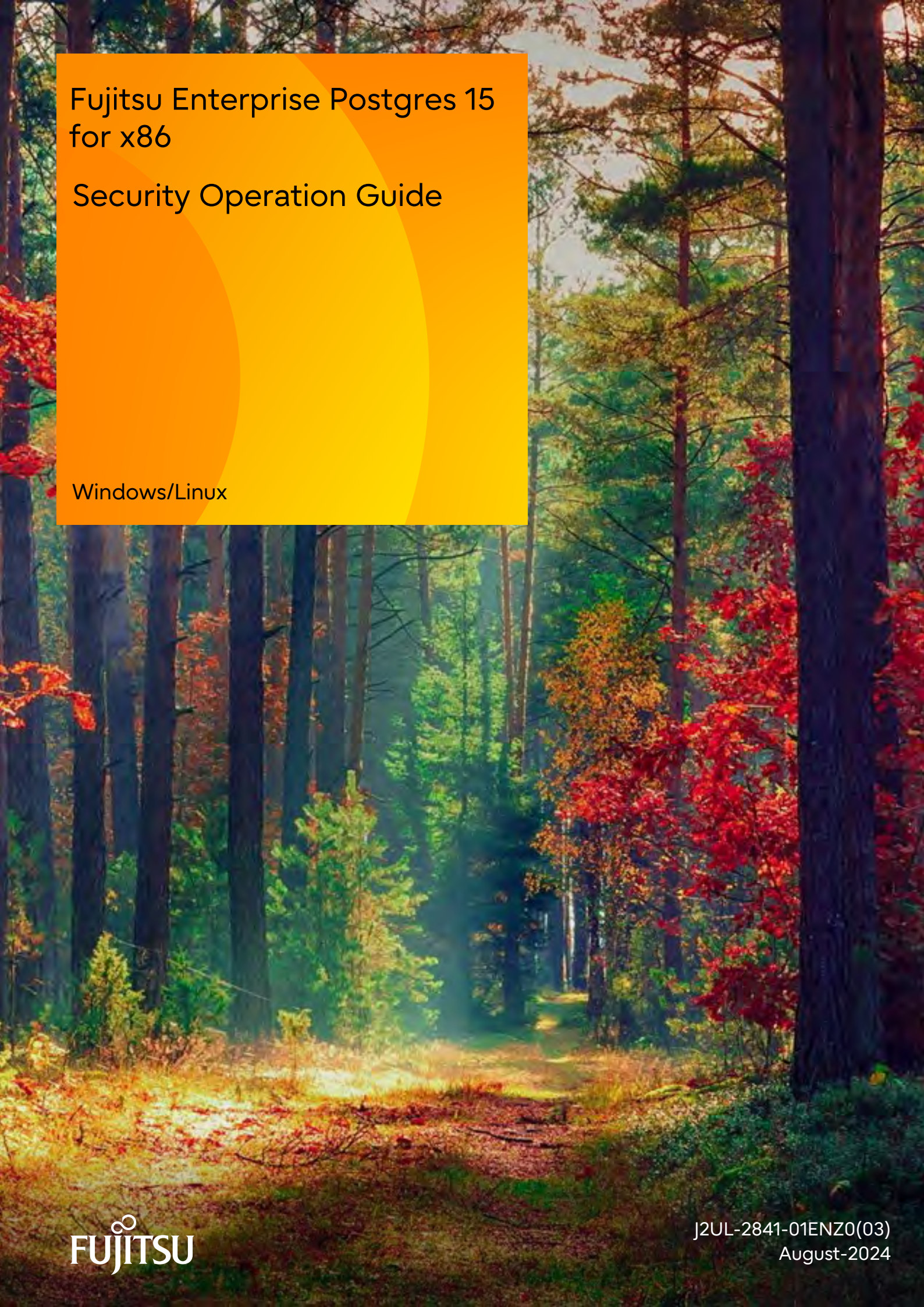
[V]
vci.control_max_workers (numerical value)....129
vci.cost_threshold (numeric)....129
vci.enable (string)....130
vci.log_query (string)....130
vci.maintenance_work_mem (numerical value)....130
vci.max_local_ros (numerical value)....130
vci.max_parallel_degree (numerical value)....130
vci.shared_work_mem (numerical value)....130

[W]
WAL and temporary files....27
WAL Mirroring Control Functions....135
WebAdmin Wallet....157

Fujitsu Enterprise Postgres 15 for x86

Security Operation Guide

Windows/Linux



FUJITSU

J2UL-2841-01ENZ0(03)
August-2024

Preface

Purpose of this document

This document describes security when building and operating a Fujitsu Software Enterprise Postgres database system.

Intended readers

This document is intended for those who are:

- Considering installing Fujitsu Enterprise Postgres
- Designing, building, and operating the security operating environment in Fujitsu Enterprise Postgres
- Accessing Fujitsu Enterprise Postgres database systems

Readers of this document are assumed to have general knowledge of:

- Business operations
- Fujitsu Enterprise Postgres
- Linux

Structure of this document

This document is structured as follows:

Chapter 1 Overview of Security

Provides an overview of the security system, and explains the security features provided by Fujitsu Enterprise Postgres.

Chapter 2 Overview of Security Operation

Provides an overview of security operation.

Chapter 3 Tasks of the Manager

Explains the tasks for security measures to be implemented by the manager.

Chapter 4 Tasks of Administrators

Explains the tasks for security measures to be implemented by administrators.

Chapter 5 Tasks of Users

Explains the tasks for security measures to be implemented by users.

Chapter 6 Audit Log Feature

Explains the audit log feature provided by Fujitsu Enterprise Postgres.

Chapter 7 Confidentiality Management

Explains the confidentiality management feature provided by Fujitsu Enterprise Postgres.

Appendix A Tables Used by Confidentiality Management Feature

Explains the tables used by the confidentiality management feature.

Appendix B System Management Functions Used by Confidentiality Management Feature

Explains the functions used by the confidentiality management feature.

References

This document contains abstracts from the following document:

- Database Security Guideline Version 2.0
  (Database Security Consortium (DBSC))

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

```
Edition 1.3: August 2024
Edition 1.2: July 2024
Edition 1.1: June 2023
Edition 1.0: April 2023
```

Copyright

Copyright 2018-2024 Fujitsu Limited

Revision History

Revision	Location	Version
Corrected the article about the audit_log_disconnections parameter.	Chapter 6 Audit Log Feature	Edition 1.3
Added scalable audit logging. This feature is available by applying the bug fix that includes the PH23619.	Chapter 6 Audit Log Feature	Edition 1.2
Feature names have been improved. Before modification : Confidential group After modification : Confidentiality group Before modification : Confidential level After modification : Confidentiality level Before modification : Confidential management role After modification : Confidentiality management role Before modification : Confidential management support After modification : Confidentiality management Before modification : Confidential matrix After modification : Confidentiality matrix Before modification : Confidential object After modification : Confidentiality object Before modification : Confidential privilege After modification : Confidentiality privilege Before modification : Confidential role After modification : Confidentiality role	Chapter 7 Confidentiality Management Appendix A Tables Used by Confidentiality Management Feature Appendix B System Management Functions Used by Confidentiality Management Feature	Edition 1.1

Contents

Chapter 1 Overview of Security....1
1.1 What is Security?....1
1.2 Security Requirements....1
1.3 Security Threats....2
1.4 Security Scope....5
1.5 Security Provided by Fujitsu Enterprise Postgres....6
1.5.1 Roles Targeted For Security....6
1.5.2 Security Features....7

Chapter 2 Overview of Security Operation....9
2.1 Security Operation Flow....9

Chapter 3 Tasks of the Manager....11
3.1 Defining Important Information and Risk Analysis....11
3.2 Formulating Account Management Policies....11
3.3 Formulating Log Retrieval Policies....11
3.4 Formulating Rules....12
3.5 Implementing Training....13
3.6 Checking the Database Management Operations....13
3.7 Periodic Diagnosis of the Status of Security Measures....13

Chapter 4 Tasks of Administrators....14
4.1 Receiving Training....14
4.2 Initial Setup....14
4.3 Authentication....15
4.3.1 Managing Accounts....15
4.3.2 Managing Passwords....16
4.3.3 Configuring Connections and Authentication....17
4.4 Access Control....17
4.5 Encryption....18
4.6 Controlling Use of External Media....18
4.7 Security Measures for Servers/Applications....18
4.8 Log Management....19
4.8.1 Retrieving Logs....19
4.8.2 Maintaining Logs....19
4.9 Detecting Unauthorized Access....20
4.10 Analyzing Logs....20

Chapter 5 Tasks of Users....21
5.1 Receiving Training....21
5.2 Managing Accounts/Passwords....21

Chapter 6 Audit Log Feature....22
6.1 Audit Log Output Modes....22
6.2 Setup....22
6.3 Setting Up the Scalable Audit Log Feature....25
6.4 pgaudit Configuration File....27
6.5 Session Audit Logging....33
6.6 Object Audit Logging....39
6.7 Database Multiplexing....41
6.7.1 Setup....41
6.7.2 Configuring Audit Log Retrieval....42
6.8 Failover....42
6.8.1 Configuring Audit Log Retrieval....42
6.9 Analyzing Audit Logs in SQL....43
6.10 Removing Setup....45

Chapter 7 Confidentiality Management ... 46
7.1 Setup ... 46
7.2 Designing Confidentiality Management ... 46
7.2.1 Designing a Confidentiality Matrix ... 46
7.2.1.1 Defining Confidentiality Levels ... 47
7.2.1.2 Defining Confidentiality Groups ... 48
7.2.1.3 Defining Confidentiality Privilege ... 49
7.2.2 Determining Confidentiality Management Roles ... 50
7.2.3 Classify Confidentiality Objects According to the Definition of Confidentiality Level ... 51
7.2.3.1 Defining Confidentiality Objects ... 51
7.2.3.2 Classify Confidentiality Objects ... 51
7.2.4 Classify Roles According to Confidentiality Group Definitions ... 51
7.3 How to Use Confidentiality Management Feature (Definition) ... 52
7.3.1 Creating a Confidentiality Management Role ... 54
7.3.2 Creating a Confidentiality Matrix ... 55
7.3.3 Adding Confidentiality Levels to the Confidentiality Matrix ... 56
7.3.4 Adding Confidentiality Groups to the Confidentiality Matrix ... 56
7.3.5 Granting Confidentiality Privileges to Confidentiality Groups ... 57
7.3.6 Adding Confidentiality Objects to Confidentiality Level ... 57
7.3.7 Adding Roles to Confidentiality Groups ... 58
7.4 How to Use Confidentiality Management Feature (Change and Deletion) ... 58
7.4.1 Renaming Confidentiality Objects ... 58
7.4.2 Renaming Roles ... 58
7.4.3 Deleting Roles ... 58
7.4.4 Changing Confidentiality Matrix ... 58
7.4.5 Deleting Confidentiality Matrix ... 58
7.4.6 Changing Confidentiality Level ... 59
7.4.7 Deleting Confidentiality Level ... 59
7.4.8 Changing Confidentiality Group ... 59
7.4.9 Deleting Confidentiality Group ... 59
7.4.10 Revoking Confidentiality Privileges ... 60
7.4.11 Removing Confidentiality Objects from Confidentiality Level ... 60
7.4.12 Removing Roles from Confidentiality Groups ... 60
7.5 Suggestions for Monitoring Methods ... 60
7.5.1 How to Detect Privilege Changes without Using Confidentiality Management feature ... 60
7.5.2 How to Check Confidentiality Objects and Roles ... 61
7.6 Backup/Restore ... 62
7.7 Removing Setup ... 63
7.8 Usage Example of Confidentiality Management ... 63

Appendix A Tables Used by Confidentiality Management Feature ... 66
A.1 pgx_confidential_matrix ... 66
A.2 pgx_confidential_level ... 66
A.3 pgx_confidential_group ... 67
A.4 pgx_confidential_privilege ... 67
A.5 pgx_confidential_object ... 68
A.6 pgx_confidential_role ... 69
A.7 pgx_confidential_policy ... 69

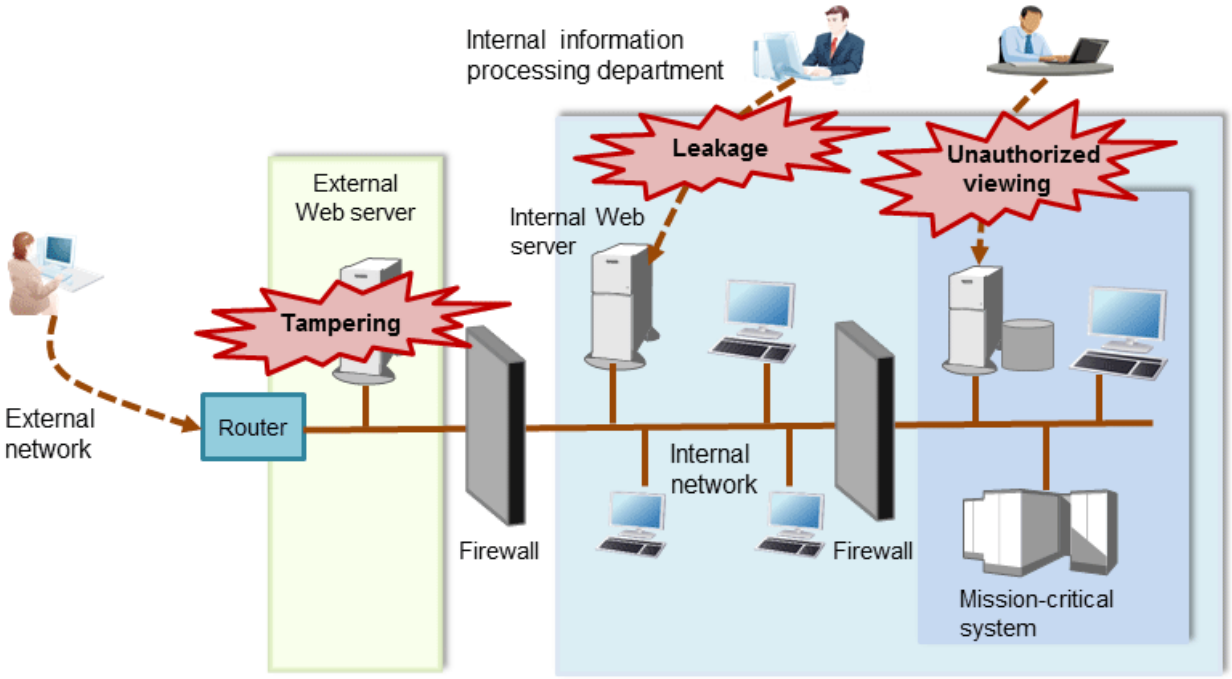
Appendix B System Management Functions Used by Confidentiality Management Feature ... 71
B.1 Confidentiality Matrix Manipulation Functions ... 71
B.2 Confidentiality Level Manipulation Functions ... 72
B.3 Confidentiality Group Manipulation Functions ... 74
B.4 Confidentiality Privilege Manipulation Functions ... 76
B.5 Confidentiality Object Manipulation Functions ... 77
B.6 Role Manipulation Functions ... 80
B.7 Functions that Support Definition Referencing and Comparison with System Catalogs ... 81

Chapter 1 Overview of Security

1.1 What is Security?

Computer security is the protection of information systems and data from risks such as leakage or tampering of information, attacks, intrusions, eavesdropping from external sources, and interference with information services. Security measures are essential for the advance prevention of security threats in order for information systems to gain trust as social infrastructure.

Figure 1.1 Security threats



The security measures in information systems can be classified as follows:

- Network
- Web
- Application
- Database
- PC

This document focuses on database security measures when using Fujitsu Enterprise Postgres.

1.2 Security Requirements

Below are the necessary security requirements for information systems.

Maintenance of security policies

A security policy clarifies the approach the company should take in relation to information assets, and the actions employees should take.

It is necessary to undertake security of information systems while maintaining security policies.

Integrated security management

Security has the aspects below. It is necessary to manage information in an integrated manner based on these aspects.

Confidentiality

Access to the information is restricted to prevent leakage of information outside of the company

Example measures: Prevention of information leakage or setup of access privileges

Integrity

Integrity is guaranteed, ensuring information does not become corrupted or tampered with

Example measures: Prevention or detection of tampering

Availability

Failure is prevented and normal operation is maintained so that information can be used when needed

Example measures: Power supply measures, system mirroring

1.3 Security Threats

A security threat is defined as something that threatens the confidentiality, integrity, and availability indicated in "1.2 Security Requirements" in respect to information assets. This includes technical threats such as accessing a database, but does not include physical destruction.

Threats are considered to be a combination of type of user who is the source of the threat, information assets that need to be protected, techniques, and unauthorized actions. For example, a threat might be a general user exploiting a database vulnerability to obtain database management information, and then tampering with that information.

When considering security measures, it is firstly necessary to clarify what kind of threats there are. A list of possible threats is shown in the table below. Refer to "Types of user" and "Information assets" for details on the definition of each type of user and information assets that should be protected.

Possible threats

Type of user	Information asset	Technique	Unauthorized action
General user Internal user System manager System developer System administrator System operator	Database management information	Eavesdropping of packets	Unauthorized acquisition (viewing) of information Unauthorized tampering or destruction (updating) of information
		Dictionary attack of passwords	
		Unauthorized acquisition of IDs/passwords through social engineering	
		Unauthorized acquisition of information through misuse of settings	
		Unauthorized acquisition of information through exploiting a database vulnerability	
		Acquisition by an unauthorized route	
General user Internal user	General database information	Acquisition by a normal route	Misuse of information that can be acquired normally (taking data outside of the company)
		SQL issued with the aim of obstructing a job	Obstructing a job (resource depletion)

Type of user	Information asset	Technique	Unauthorized action
General user Internal user	General database information	Eavesdropping of packets	Unauthorized tampering or destruction (updating) of information
		Dictionary attack of passwords	
		Unauthorized acquisition of IDs/passwords through social engineering	
		Unauthorized acquisition of information through exploiting configuration errors	
		Unauthorized acquisition of information through exploiting a database vulnerability	
		Acquisition by an unauthorized route	
System manager System developer System administrator System operator	General database information	Eavesdropping of packets	Unauthorized acquisition (viewing) of information Unauthorized tampering or destruction (updating) of information
		Dictionary attack of passwords	
		Unauthorized acquisition of IDs/passwords through social engineering	
		Unauthorized acquisition of information through exploiting configuration errors	
		Unauthorized acquisition of information through exploiting a database vulnerability	
		Acquisition by an unauthorized route	
System developer	Database management information	Creation of a backdoor	Unauthorized acquisition (viewing) of information Unauthorized tampering or destruction (updating) of information
	General database information		
System manager System administrator	Database management information	Unauthorized acquisition of information by creating an unauthorized database administrator account	Unauthorized acquisition (viewing) of information Unauthorized tampering or destruction (updating) of information
	General database information		
System manager System operator	Database management information	Unauthorized acquisition of information by tampering with database-related files	Unauthorized acquisition (viewing) of information

Type of user	Information asset	Technique	Unauthorized action
	General database information	(definition file, physical file, and so on)	Unauthorized tampering or destruction (updating) of information
Database administrator	Database management information	Misuse of information (taking information outside of the company) after obtaining it through the normal route	Misuse of information that can be acquired normally (taking information outside of the company)
		Unauthorized use of IDs/ passwords from the management information	Tampering with or destroying information that can be acquired
		Unauthorized acquisition of information by tampering with management information	
		SQL issued with the aim of obstructing a job	Obstructing a job (resource depletion)
	General database information	Eavesdropping of packets	Unauthorized acquisition (viewing) of information Unauthorized tampering or destruction (updating) of information
		Misuse of information (taking information outside of the company) after obtaining it through an unauthorized route	
Database operator	Database management information	Eavesdropping of packets	Unauthorized acquisition (viewing) of information Unauthorized tampering or destruction (updating) of information
		Dictionary attack of passwords	
		Unauthorized acquisition of IDs/passwords through social engineering	
		Unauthorized acquisition of information by exploiting configuration errors	
		Unauthorized acquisition of information through exploiting a database vulnerability	
		Acquisition by an unauthorized route	
	General database information	Acquisition by a normal route	Misuse of information that can be acquired normally (taking data outside of the company)
		SQL issued with the aim of obstructing a job	Obstructing a job (resource depletion)

Types of user

In database security, the persons involved with databases and their roles are defined below.

Type of user	Role
System manager	Manages developers, administrators, and operators
System developer	Builds the network around the database server Builds the database server
System administrator	Operates devices of the surrounding database network Operates the database server
System operator	Operates the surrounding database network
Database administrator	Builds the database system Operates the database system
Database operator	Performs business operations
Internal user	End user inside the company
General user	End user outside the company

Information assets

In database security, it is necessary to protect the information assets to be stored on the database server.

Such assets are defined below.

Database management information

- Database configuration information (system catalog, user ID/password, and so on)
- Database logs (such as access logs)

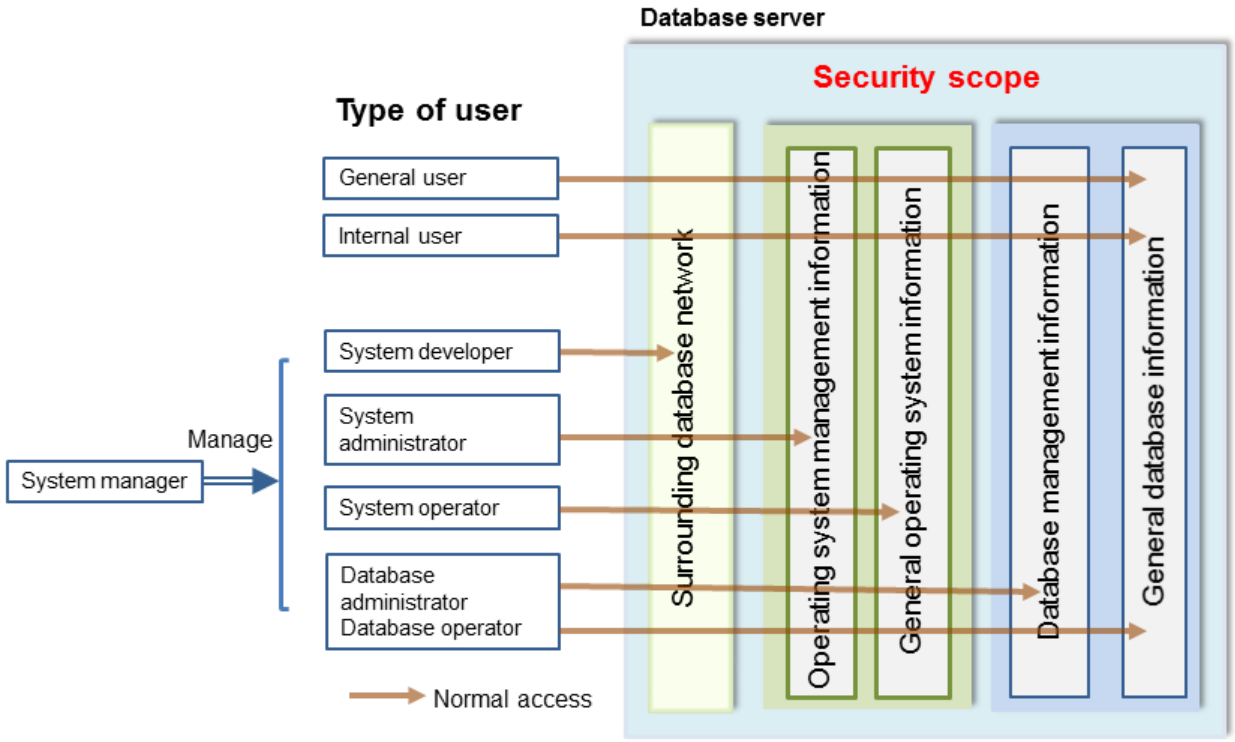
General database information

- Job data
- Applications

1.4 Security Scope

In database systems, both the database server and the surrounding database network are part of the security scope. It is necessary to clarify the extent of the security scope that each type of user is involved with, and consider security measures for the same.

The relationship of the security scope and the types of user is shown below.



1.5 Security Provided by Fujitsu Enterprise Postgres

Fujitsu Enterprise Postgres provides security features that satisfy the security requirements indicated in "1.2 Security Requirements".

This section describes security provided by Fujitsu Enterprise Postgres.

1.5.1 Roles Targeted For Security

In Fujitsu Enterprise Postgres database systems, the roles targeted in relation to security are "Manager", "Administrator", and "User". In order to build a robust security system, it is necessary to put security measures in place for each role.

The roles targeted for security and the mapping of Types of user indicated in "1.3 Security Threats" are shown in the table below.

Role targeted for security	Type of user
Manager	System manager
Administrator	System developer
	System administrator
	System operator
	Database administrator
	Database operator
User	General user
	Internal user

Manager

The manager establishes a security policy and decides on an operations policy for the organization as a whole.

Refer to "Chapter 3 Tasks of the Manager" for details.

Administrator

Administrators design, build and operate a system. While doing this, the administrators must implement the security measures in accordance with the security policy established by the manager.

Refer to "Chapter 4 Tasks of Administrators" for details.

User

A user is a person other than the manager or an administrator who accesses a database. There may be any number of users. It is necessary for users to be registered in the database system, and that access to the database is restricted according to the access privileges.

Refer to "Chapter 5 Tasks of Users" for details.

1.5.2 Security Features

Fujitsu Enterprise Postgres provides the following security features:

- Authentication
- Access control
- Encryption
- Audit log
- Data masking

This section describes each of these features.

Authentication

The databases that can be accessed can be restricted by authenticating the database users who access the database. Additionally, authentication of the server can be performed to prevent spoofing of the database server.

Refer to "Client Authentication" in "Server Administration" in the PostgreSQL Documentation for details on authentication.

Refer to "Secure TCP/IP Connections with SSL" in "Server Setup and Operation" in the PostgreSQL Documentation for details on server authentication.

Access control

Database objects can only be used by the object creator or database user who was specified as the owner when the object was created (both persons are hereinafter referred to as "owner"), or superuser, when objects are in their initial state. By having the object owner or superuser control access privileges for database users, it is possible to control what kind of tables the database users who connect to the database can access, and what kind of operations they can perform.

Fujitsu Enterprise Postgres provides security management support features that support the design and operation of access control. For details, refer to "Chapter 7 Confidentiality Management".

Refer to "Privileges" in "The SQL Language" in the PostgreSQL Documentation for details on object access control.

Encryption

Fujitsu Enterprise Postgres provides a transparent data encryption feature that satisfies the requirements below.

- Confidential information can be changed into an unidentifiable state.
- The encryption key and data are managed separately.
- The encryption key is replaced at regular intervals.

Also, confidential data should not be operated without encryption. Fujitsu Enterprise Postgres provides security management support features to help prevent this. For details, refer to "Security Management Support".

PostgreSQL provides an encryption feature called "pgcrypto" that can also be used in Fujitsu Enterprise Postgres, however, it is recommended to use the transparent data encryption features because it will otherwise be necessary to modify the applications that consider encryption. Refer to "Protecting Storage Data Using Transparent Data Encryption" in the Operation Guide for details.

Additionally, if communication data transferred between a client and a server contains confidential information, it is necessary to encrypt the communication data to protect it against threats, such as eavesdropping on the network.

Refer to "Configuring Secure Communication Using Secure Sockets Layer" in the Operation Guide for details on encryption of communication data.

Audit log

A feature that addresses threats such as misuse of administrator privileges, unauthorized access to a database by a user, and other such threats. Information for tracing the processing of administrators and users is retrieved and stored as an audit log.

By periodically viewing and monitoring audit logs, the administrators can detect events that are impacting on the system in some way, or are depleting system resources as a result of incorrect operations by users, and can take appropriate measures to prevent information leakages or system failures in advance.

Refer to "Chapter 6 Audit Log Feature" for details.

Data masking

A feature that changes part of the data to make it available for reference in response to queries issued by an application.

For example, for a query of employee data, digits except the last four digits of an eight-digit employee number can be changed to "*" so that it can be used for reference without exposing the actual data.

Specifically, the data changed by the data masking feature can be transferred to a test database so that users who perform testing or development can reference the data. During testing, it is desirable to use the data that will be used on a production environment database. However, actual production data should not be used as is for testing because of the risk of leakage of confidential data. This feature enables data that is similar to actual production data to be safely used in test and development environments.

Refer to "Data Masking" in the Operation Guide for details on data masking.

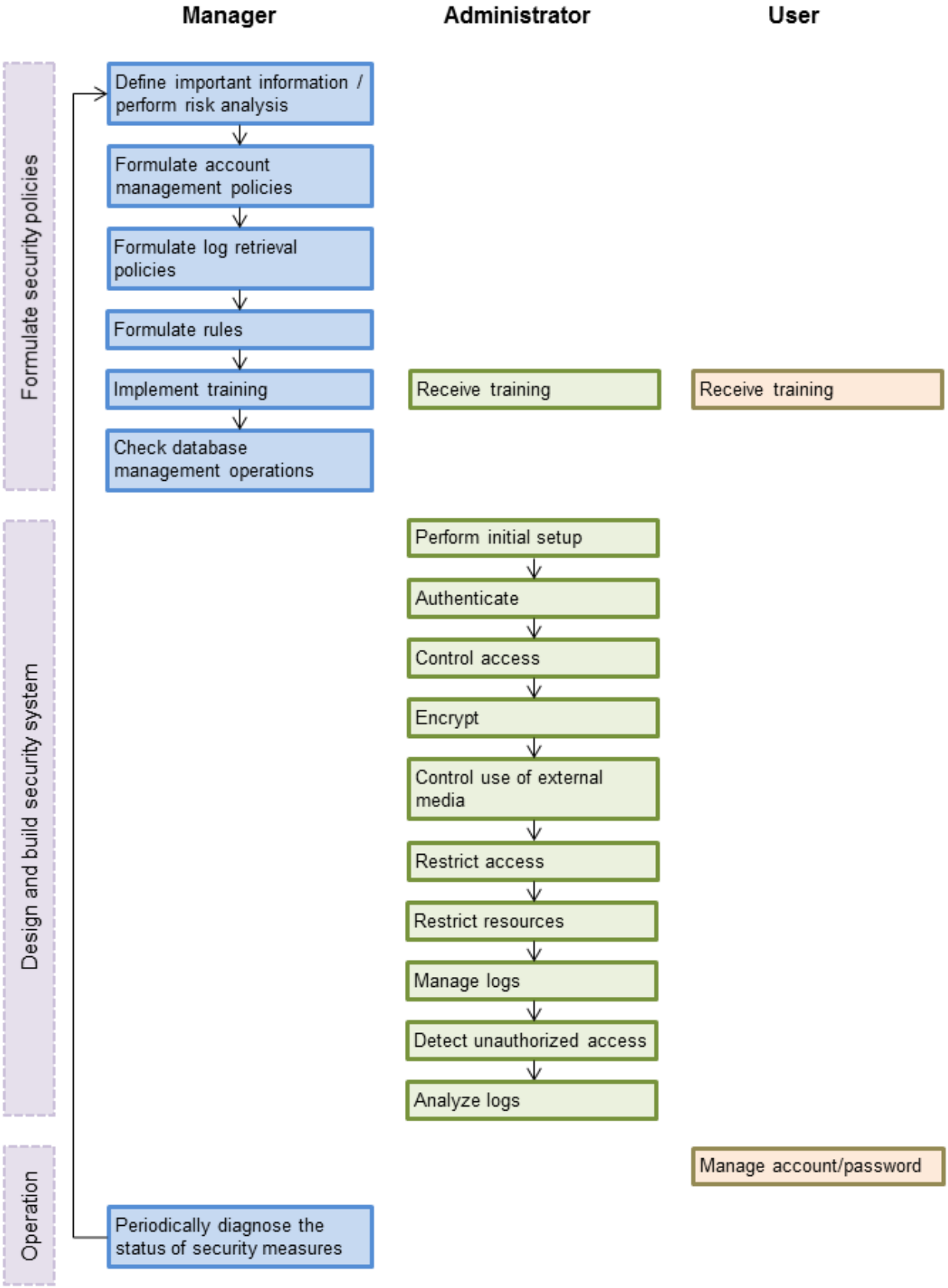
Chapter 2 Overview of Security Operation

2.1 Security Operation Flow

This section shows the flow of work when building a security environment and performing security operation in Fujitsu Enterprise Postgres.

When performing security operation, there are technical measures to be implemented to address security threats by equipping the system with security features, and manual work, such as the implementation of security guidelines, a training system, and the establishment of usage rules.

Figure 2.1 Security operation flow



Chapter 3 Tasks of the Manager

The manager formulates security policies, which become guidelines for security measures.

3.1 Defining Important Information and Risk Analysis

Before formulating security policies, define important information and perform risk analysis. Based on the importance of the information and the result of risk analysis, decide what kind of security measures to put in place.

In defining the important information, identify what should be protected and classify it by importance in order to effectively implement the security measures. Information that should be protected includes "database management information" and "general database information", as indicated in "Information assets". Examples of information classifications are "personal information" and "confidential information".

In the risk analysis, refer to "Possible threats" to identify threats that may arise, and analyze the risks in respect to such threats.

Additionally, by performing a risk analysis once annually as a guide, it is possible to identify threats that may adversely impact the business and related vulnerabilities.

3.2 Formulating Account Management Policies

In formulating an account management policy, implement the following and document the formulated policy.

Organize system users and roles

Identify the necessary roles of the relevant system based on "Types of user". Additionally, organize personnel for each role.

Organize accounts

Organize accounts with the appropriate privileges for each role, and decide on account policies.

- Database administrator account
  - Organize separate accounts for database administrators and database operators
  - Ensure that the database administrator account can only be used by specific persons
  - Perform tasks that do not require database administrator privileges using a separate account without database administrator privileges
- General account

  Create an account for general users by application usage.

Review account management policy

Review the accounts in order to effectively implement security measures.

- Regularly check the accounts mentioned above and their privileges, and determine if they are still appropriate
- If there have been system or operational changes, review the accounts and privileges
- If unsuitable accounts and privileges are discovered, modify them as required

3.3 Formulating Log Retrieval Policies

In formulating a log retrieval policy, implement the following and document the formulated policy.

Organize the purpose of log retrieval

To clarify what logs will be retrieved for, define their reason for retrieval.

Examples of the purpose might include, "To use for investigation in the event of unauthorized access", and "To submit to investigating authorities as evidence if any issues arise".

Decide on the types of logs to be retrieved

In order to retrieve appropriate logs, organize the types of logs that can be retrieved in the target system, and decide on the logs to be retrieved.

Examples of log types are "operating system logs", "application run logs", and "database audit logs".

Organize log retrieval target access

In order to decide on access for log retrieval targets, organize what kind of access will take place.

For example, the following access is possible:

- Access related to important information
  - Access to personal information, confidential information, and database management information
  - Access outside of business hours
  - Login
  - Specific SQL
- Access suspected to be unauthorized
  - Large amount of search access
  - Access from different locations
  - Access outside of business hours

Decide on the log retrieval content

In order to effectively use retrieved logs, organize the required content as a log, and decide on the retrieval content.

For example, the following output content is possible:

- When (time)
- Who (database account, application user)
- What (object ID, table name)
- Where from (machine name, IP address)
- How (SQL type, SQL statement)
- Execution result (success/fail)

Formulate log maintenance policy

In order to use the logs as purposed, formulate the log maintenance policy.

For each log, define its location, storage medium, retention period, access control, and so on.

3.4 Formulating Rules

Formulate the rules that will become the standard for security measures of the target system. Additionally, prescribe penalties for security violations. For example, formulate rules and penalties as below:

- Rules
  - Applying security patches and update programs
  - Prohibiting unauthorized acquisition of information from the database
  - Prohibiting the saving of acquired information to media that is not permitted for use
- Penalties
  - Prescribe penalties in the company's employment policies and procedures

- Set fines

3.5 Implementing Training

In order to have administrators and users recognize the importance and necessity of information security, and to prevent unauthorized access due to operational omissions and mistakes, implement and promote security-related training for administrators and users.

For example, implement promotion of security policies, formulation of training schedules, and formulation of training materials.

3.6 Checking the Database Management Operations

In order to prevent operational errors and unauthorized actions by administrators, implement the measures below:

- Always collect the latest information on security incidents and vulnerabilities related to databases
- Implement management operations only after providing advance notice
- Retain records of management operations

3.7 Periodic Diagnosis of the Status of Security Measures

In order to check if the security measures are effective, periodically diagnose if the security measures have been put in place appropriately based on the security threats.

Additionally, evaluate if the current security measures and policies are effective for the threats and vulnerabilities, and if there are any issues, review the security policies and security measures.

Chapter 4 Tasks of Administrators

Administrators perform the actions below as security measures when designing, building, and operating the system in accordance with the security policies formulated by the manager.

Preparation

- Implement training

Measures to protect against unauthorized behavior

- Perform initial setup
- Authenticate
- Control access
- Encrypt
- Control use of external media
- Restrict access
- Restrict resources

Measures to detect and trace unauthorized behavior

- Manage logs
- Detect unauthorized access
- Analyze logs

4.1 Receiving Training

Administrators receive security-related training in accordance with the training schedule formulated by the manager. Additionally, administrators instruct users to receive training.

4.2 Initial Setup

To minimize database vulnerabilities and the possibility of unauthorized access, implement the security measures below in the initial stage of system building. Additionally, configure the database server so that it primarily operates the database system only.

Making the server more robust

Configure the operating system and network to prevent intrusion into or destruction of a database server, so that the system operates on a secure server.

- Remove unnecessary features or services on the operating system
- Enable only the necessary protocols
- Implement the security features for services, protocols, and daemons considered to have a relatively low security level, such as file sharing and FTP

Installing the latest version

Always download and apply the latest patches in order to reflect the latest security measures.

Installing the minimum necessary features

Install only the necessary features in order to prevent unauthorized use of the system.

Additionally, delete or disable features and services that will not be used.

Changing the port

To prevent unauthorized use of the system, change the default port that is set during installation.

Point

Specify the port during setup of Fujitsu Enterprise Postgres. Refer to the Installation and Setup Guide for Server for details.

Access restrictions for communication features

To prevent unauthorized use of the system using the communication features, implement access restrictions for communication features.

Settings for prohibiting the access path to database configuration files

To prevent database destruction, implement the measures below:

- Restrict users who are permitted to access database configuration files, and periodically review the permissions
- Allow only administrators to access table or definition scripts

Restrictions on the access path to the database

To prevent unauthorized use or operating errors for the database, restrict the distribution range of applications used to access the database only to devices used by users who are permitted access.

Dealing with unauthorized programs

To prevent unauthorized intrusions into a system through a backdoor, such as by tampering with the program source code of an application, document the author of the program to be run and perform checking and testing so that the program will not be tampered with. Additionally, employ safe coding techniques so that issues with general coding vulnerabilities can be addressed.

System security settings

In cases where it is clear that the system security settings will impact security, set reliable security settings in the initial setup stage, such as setting appropriate security parameters.

4.3 Authentication

When accessing a database, authentication must always be performed in order to prevent tampering or information leakage from spoofing by a malicious user.

Password authentication is used when logging on to a database, and the account and password used for authentication are to be strictly managed by administrators.

Additionally, authentication must also be implemented reliably for connections to a database from clients, so that only permitted users can access the database.

4.3.1 Managing Accounts

For account management, perform the actions below.

Create the required accounts

To prevent unauthorized use of accounts, such as spoofing, implement the measures below when creating an account:

- Select the required account
- Specify the user privileges
- Create database administrator accounts and general user accounts separately according to the privileges

Point

Accounts are created using the CREATE ROLE statement. Refer to "CREATE ROLE" in the PostgreSQL Documentation for details.

Delete unnecessary accounts

Remove accounts not used on a daily basis, such as unused accounts and accounts not needed for operations that are created by default during product installation.

Point

Accounts are deleted using the DROP ROLE statement. Refer to "DROP ROLE" in the PostgreSQL Documentation for details.

Set up account lockout

The usage frequency of accounts is to be checked periodically, and if there are any accounts that have not been used for a long period, lock those accounts. Set a limit for failed login attempts, and if this limit is exceeded, lock the account. Additionally, set the period until a locked account is reenabled.

Point

Account locking can be performed by using LDAP authentication. Refer to "LDAP Authentication" in the PostgreSQL Documentation for details.

Manage database administrator accounts

Manage database administrator accounts in accordance with the account management policy formulated by the manager.

Manage development environment and production environment accounts

To prevent unauthorized use of accounts used in a development environment, delete accounts used in the development environment before operation starts in the production environment. In cases where it is unavoidable to use an account used in the development environment in the production environment, use different passwords in each environment.

Set up a temporary use account

If a temporary user will use the system, either provide a shared account with a temporary password for each use, or create a temporary account.

4.3.2 Managing Passwords

Manage passwords as below.

Make strong passwords

The use of account passwords that can easily be guessed by others, such as a password that matches the ID, or the default password provided during installation, is prohibited. Set a complex and strong password.

Change passwords regularly

Change passwords regularly to prevent others from accessing the account in case the password is obtained by unauthorized means. Additionally, configure the settings to force a password change when prompted after the first use.

Set the password expiry period

To encourage regular changing of passwords, set a password expiry period.

Point

Password setting and changing is specified using the CERATE ROLE statement or ALTER ROLE statement. Refer to "CREATE ROLE" and "ALTER ROLE" in the PostgreSQL Documentation for details.

Additionally, by using passwordcheck and LDAP authentication, the actions below can be performed:

- The default password set during installation can be changed
- The password expiry period can be set

- The number and types of characters used for the password can be checked

Refer to "passwordcheck" and "LDAP Authentication" in the PostgreSQL Documentation for details.

4.3.3 Configuring Connections and Authentication

Configure connections and authentication so that the database can only be accessed by permitted users.

Point

Client authentication is configured in pg_hba.conf. Refer to "Client Authentication" in the PostgreSQL Documentation for details.

4.4 Access Control

If appropriate access privileges are not set for administrators and users, security incidents may occur, such as information leakage resulting from access to information by an unauthorized person. To minimize such incidents, it is necessary to implement the security measures below for the access privileges and perform rule-based access control.

Point

Notes when setting access privileges

- The creation of a special account that allows granting of privileges to all users is prohibited
- The creation of a general account that allows access to general information such as operations data is prohibited

Identifying the database access requirements

To set the appropriate access privileges for each usage purpose for the database, follow the procedure below to identify the access requirements:

1. Classify the usage purpose of the account, such as "For database management", "For object management", and "For data access".
2. Classify the required privileges for each usage purpose, such as "By feature" and "By object".
3. Categorize the accounts based on each privilege.
4. Identify the minimum necessary range of data and minimum necessary access content (view, update, create, delete) to be accessed for each categorized account, and decide on the database access requirements.

Setting the access privileges

Assign the minimum necessary privileges based on the database access requirements for each categorized account. Additionally, restrict accounts when assigning administrator privileges.

Reviewing access privileges

To reflect changes in access requirements in the system, periodically review the access privileges and check if there are any access privileges that are no longer needed. If any unnecessary access privileges have been set, promptly modify the access privileges.

Point

Access privileges are set using the GRANT statement or REVOKE statement. Refer to "GRANT" and "REVOKE" in the PostgreSQL Documentation for details.

4.5 Encryption

To prevent unauthorized usage of data in the event information leakage occurs due to data theft, eavesdropping of communication, and other such activities, implement the encryption measures below.

Encrypt communication

To protect data from eavesdropping over the network between a database server and clients, use the encryption feature to encrypt communications.

Refer to "Configuring Secure Communication Using Secure Sockets Layer" in the Operation Guide for details.

Encrypt data

To protect data from theft, use the encryption feature to encrypt the data. The data below is targeted for encryption:

- Data to be stored on the database
- Backup data
- Data files

Refer to "Protecting Storage Data Using Transparent Data Encryption" in the Operation Guide for details.

Manage encryption keys

Restrict the persons who can access the encryption key to a minimum number of database administrators.

Additionally, to ensure the encrypted information will not be easily decrypted, create a mechanism for appropriately managing the encryption key for the entire life cycle (generation, distribution, saving, and disposal), and strictly manage the encryption key.

Refer to "Configuring Secure Communication Using Secure Sockets Layer" and "Protecting Storage Data Using Transparent Data Encryption" in the Operation Guide for details.

4.6 Controlling Use of External Media

Information leakage can be prevented by controlling use of external media (such as CD/DVD, USB drive, and external hard disk) and PCs that are connected to the database, and restricting the removal of data from the database.

Restricting connection of external media

Remove external media and printers that will not be used in operations, and restrict connection of external media to which information may be written.

Restricting use of external media

Restrict connections for external media and printers to control the writing of information to these devices.

Controlling use of connected PCs

Prevent leakage of information from PCs connected to the database:

- Limit connections of external media to PCs
- Implement security measures to make the PC robust
- Implement individual authentication for access from the PC
- Manage the installed software and monitor the software usage status
- Limit connections to printers

4.7 Security Measures for Servers/Applications

An even more robust security system can be achieved by strengthening security for servers and applications in addition to the security measures for databases. Implement the security measures below for servers and applications:

Restrict access

Implement the measures below and restrict access to the database server:

- Install the database server inside the firewall to prevent direct access to the database server from many unspecified PCs.
- In the local network, implement measures such as using the router to restrict IP addresses, and restrict PCs and segments that can directly access the database server.

Restrict resources

Restrict excessive use of CPU resources by general users to prevent the disruption of service and extraction of large amounts of data.

4.8 Log Management

Logs are a feature that addresses threats such as misuse of administrator privileges, and unauthorized access to a database by a user. Information for investigating/tracing processes and operations performed for the database is retrieved and managed as logs for identifying the cause in the event information leakage or unauthorized access occurs.

Fujitsu Enterprise Postgres provides the audit log feature for retrieving and managing logs.

Refer to "Chapter 6 Audit Log Feature" for details.

This section describes the information that should be obtained as logs and how to maintain logs, as a measure for managing information leakage and unauthorized access.

4.8.1 Retrieving Logs

The audit logs below are retrieved in accordance with the log retrieval policy formulated by the manager.

Login information

Retrieves logs during login and logout.

Database access information (view/update)

Retrieves all access relating to the information below:

- General database information (such as personal information and confidential information used in the business)
- Database management information (system catalog, user ID/password, and so on)

Changed information of database objects

Retrieves logs related to creating, changing, and deleting database objects such as database accounts and tables.

Operation logs for audit logs

To prevent suppression of retrieved audit logs, operations such as initialization of audit logs, and stoppage of the audit log feature are retrieved as logs.

4.8.2 Maintaining Logs

Logs are maintained in accordance with the log retrieval policy formulated by the manager.

Storing logs

Perform the actions below and store logs securely so that the retrieved logs will not be updated by others:

- Save logs to external media, and store the external media in a secure location, such as lockable storage
- Restrict the viewing of logs to administrators only, and set access restrictions for logs, such as not assigning update rights
- Decide on the log retention period, with consideration to cases where investigation tracing back to the time of discovery of an issue is required

Preventing tampering of logs

Implement measures to prevent tampering of logs, such as retaining multiple copies of logs and using storage that cannot be rewritten.

Encrypting logs

Encrypt logs so that logs are not easily viewed.

4.9 Detecting Unauthorized Access

To address unauthorized access, it is necessary to establish a mechanism for detecting unauthorized access to databases and monitor access.

Communicating unauthorized access

Create a mechanism that notifies of detected unauthorized access, such as notifying the manager and the administrator, if an account lock occurs due to the limit for failed login attempts being exceeded.

Checking access times

Create a mechanism that can check for suspicious access to the information below outside of normal access hours, together with implementing measures to address such access.

Detecting access to database management information

- Monitor logs and detect access during timeframes that have not been applied for
- In the event a request for access permission outside of normal access hours is made, the log is checked for any discrepancies in the requested content and work result

Detecting access to general database information

- Decide on the timeframes during which access to the database is permitted for each general account
- Detect access outside of normal access hours from session information logs

Checking the connection source where access is not permitted

To detect access from connection sources that are not permitted, define the sources from where access is permitted, and detect access from connection sources that are not permitted.

Define the access patterns (connection source, operating system user and account) of database administrator accounts and general accounts, and check for access outside of these patterns.

4.10 Analyzing Logs

Create a mechanism that analyzes logs to detect unauthorized behavior in cases where information leakage, unauthorized access, or other such activity, is suspected. Analyses should include those shown below.

Periodic analysis of session information

Analyze session information of logs from the perspectives below to detect unauthorized logins:

- Trend of sessions with a large number of failed login attempts
- Trend of sessions with accounts that are logged in for long periods of time
- Trend of sessions in which a large amount of resources are used

Periodic analysis of database access information

Analyze SQL statements from the perspectives below to detect unauthorized access to the database:

- Trend of SQL being executed over a long period of time
- Trend of SQL using a large amount of resources

Chapter 5 Tasks of Users

The user performs the actions below as security measures when using the system.

5.1 Receiving Training

The user must receive security-related training as instructed by the manager or the administrator to learn about security. By having users with a common awareness relating to security, an even more stable security system can be established.

5.2 Managing Accounts/Passwords

Users can use the database system by using the account and password provided by the administrator. At such times, the user is to implement the measures below so that the account and password are not misused by others:

- Be responsible for managing the ID and password in a manner that ensures the account does not become locked during login
- Change the password regularly
- Comply with the expiry period set for the password by promptly changing the password when it is about to expire

Chapter 6 Audit Log Feature

In PostgreSQL, logs output as server logs can be used as audit logs by using the log output feature. There are, however, logs that cannot be analyzed properly, such as SQL runtime logs, which do not output the schema name. Additionally, because the output conditions cannot be specified in detail, log volumes can be large, which may lead to deterioration in performance.

The audit log feature of Fujitsu Enterprise Postgres enables retrieval of details relating to database access as an audit log by extending the feature to pgaudit. Additionally, audit logs can be output to a dedicated log file or server log. This enables efficient and accurate log monitoring.

The scalable audit log feature enables multiple output mechanisms to capture audit logs without performance degradation, even for systems with high application multiplicity and high output volumes.

The scalable audit log feature is available by applying the bug fix that includes the PH23619.

If you want to take advantage of the scalable audit log feature, refer to "6.2 Setup" and "6.3 Setting Up the Scalable Audit Log Feature.

Note

The audit log feature cannot be used if PostgreSQL is running in single-user mode.

6.1 Audit Log Output Modes

In pgaudit, the two types of audit log below can be output.

Session Audit Logging

Session Audit Logging outputs information related to SQL executed in backend processes (processes generated when connection requests are received from clients), information related to starting and connecting databases, and information related to errors, as a log. In Session Audit Logging, by specifying the log output conditions and filtering the logs to be output, performance degradation due to outputting large volumes of logs can be prevented.

Refer to "6.5 Session Audit Logging" for details.

Object Audit Logging

When SELECT, INSERT, UPDATE, and DELETE are executed for specific objects (tables, columns), Object Audit Logging outputs these as a log. TRUNCATE is not supported. Object Audit Logging outputs object operations for which privileges have been assigned to specified roles, as a log. Object Audit Logging can control log output at an even finer level of granularity than Session Audit Logging.

Refer to "6.6 Object Audit Logging" for details.

Information

Depending on the application or command, Fujitsu Enterprise Postgres may execute SQL internally and the audit logs may be retrieved.

Also, the audit logs of multiple SQLs with the same statement ID may be retrieved. This is because before the user executes the SQL, another SQL is executed internally by Fujitsu Enterprise Postgres.

6.2 Setup

This section describes the setup method of pgaudit.

1. Copy the pgaudit files

   As superuser, run the following command. Note that "<*x*>" in paths indicates the product version.

```
$ su -
Password:******
# cp -r /opt/fsepv<x>server64/OSS/pgaudit/* /opt/fsepv<x>server64
```

W

Open a command prompt as administrator privileges, run the following command. Note that "<*x*>" in paths indicates the product version.

```
> xcopy /E "c:\Program Files\Fujitsu\fsepv<x>server64\OSS\pgaudit\*"  "c:\Program
Files\Fujitsu\fsepv<x>server64"
```

2. Create the pgaudit configuration file

   Create the pgaudit configuration file, which describes the information required for pgaudit actions. Create the file using the same encoding as used for the database.

   In addition, set write permissions for the database administrator only in the pgaudit configuration file so that policies related to the audit log are not viewed by unintended users.

   Refer to "6.4 pgaudit Configuration File" for details.

   Note

   Do not define the rule section in the pgaudit configuration file at this point.

   Example of a pgaudit configuration file

   ```
   [output]
   logger = 'auditlog'
   ```

3. Configure postgresql.conf

   Configure the parameters below in postgresql.conf to use audit logs:

   shared_preload_libraries

   Specify "pgaudit".

   pgaudit.config_file

   Specify the deployment destination path of the pgaudit configuration file.

   If a relative path is specified, the path will be relative to the data storage directory.

   log_replication_commands

   Specify "on".

   log_min_messages

   Check if "ERROR" or higher has been specified.

   If outputting an audit log to a server log ("serverlog" is specified in the logger parameter of the pgaudit configuration file), check the parameters below relating to server logs.

   logging_collector

   Check if "on" has been specified.

   log_destination

   Check if "stderr" has been specified.

L

   log_file_mode

   Check if the server log permissions are appropriate, so that only the permitted persons can access it.

Information

The default for the log_file_mode parameter is 0600, which only allows the database administrator to have access.

For example, to permit other members of the group to which the database administrator belongs to view the audit logs, specify 0640 for log_file_mode.

Example

```
log_file_mode = 0640
```

The database administrator can also be prevented from viewing audit logs by specifying 0000. However, write privileges are assigned for outputting logs.

If outputting an audit log to a dedicated log file ("auditlog" is specified in the logger parameter of the pgaudit configuration file), check the parameter below.

max_worker_processes

If the max_worker_processes parameter has been set, add 1 to the specified value.

If you want to take advantage of the scalable audit log feature, refer to "6.3 Setting Up the Scalable Audit Log Feature".

See

Refer to "Error Reporting and Logging" in the PostgreSQL Documentation for details on server logs.

If using database multiplexing, refer to "6.7 Database Multiplexing" for details.

Example of postgresql.conf

In the example below, only the parameters that need to be configured when using the audit log feature are described.

```
shared_preload_libraries = 'pgaudit'
pgaudit.config_file = 'pgaudit.conf'
log_replication_commands = on
log_min_messages = WARNING
```

4. Start the instance

   Start the instance and check if the message below is output.

```
LOG: pgaudit extension initialized
```

5. Create the pgaudit extension

   Execute CREATE EXTENSION to create the pgaudit extension.

```
$ psql
=# CREATE EXTENSION pgaudit;
=# \dx
                  List of installed extensions
Name    | Version |   Schema   |     Description
--------+---------+------------+---------------------------------
pgaudit | 1.0     | public     | provides auditing functionality
plpgsql | 1.0     | pg_catalog | PL/pgSQL procedural language
(2 rows)
```

6. Configure the parameters in the pgaudit configuration file

   Add or change the parameters in the pgaudit configuration file as required.

   Refer to "6.4 pgaudit Configuration File" for details.

7. Restart the instance

   Restart the instance to apply the changes to the pgaudit configuration file. After restarting, check if the changes have been reflected correctly.

Linux

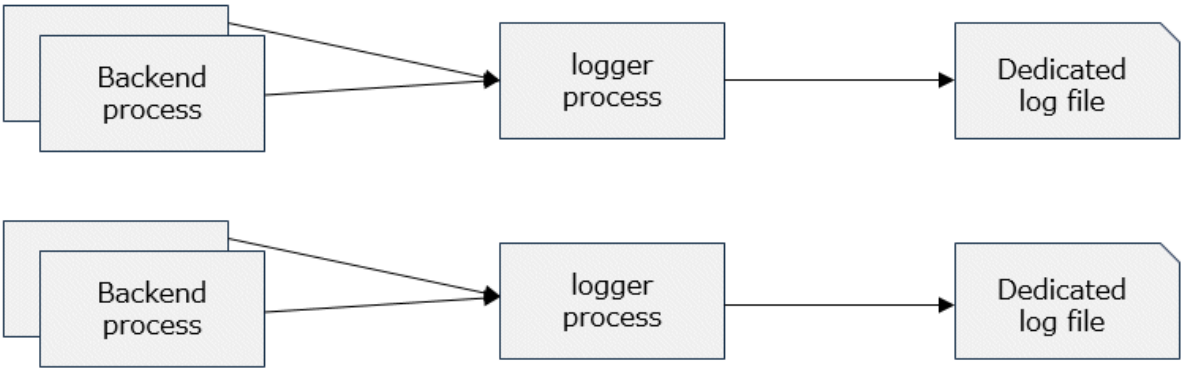
```
LOG:  log_catalog = 1
LOG:  log_level_string =
LOG:  log_level = 15
LOG:  log_parameter = 0
LOG:  log_statement_once = 0
LOG:  role =
LOG:  logger = auditlog
LOG:  log_directory = pgaudit_log
LOG:  log_filename = pgaudit-%Y-%m-%d_%H%M%S.log
LOG:  log_file_mode = 0600
LOG:  log_rotation_age = 1440
LOG:  log_rotation_size = 10240
LOG:  log_truncate_on_rotation = 0
LOG:  fifo_directory = /tmp
LOG:  Rule 0
LOG:  pgaudit extension initialized
```

Windows

```
LOG:  log_catalog = 1
LOG:  log_level_string =
LOG:  log_level = 15
LOG:  log_parameter = 0
LOG:  log_statement_once = 0
LOG:  role =
LOG:  logger = auditlog
LOG:  log_directory = pgaudit_log
LOG:  log_filename = pgaudit-%Y-%m-%d_%H%M%S.log
LOG:  log_file_mode = 0600
LOG:  log_rotation_age = 1440
LOG:  log_rotation_size = 10240
LOG:  log_truncate_on_rotation = 0
LOG:  Rule 0
LOG:  pgaudit extension initialized
```

6.3 Setting Up the Scalable Audit Log Feature

With normal audit log feature, there is only one logger process and only one dedicated log file.
Therefore, the load of audit logs, including logs output by all backend processes, in the instance concentrates to the logger process and dedicated log file.
The scalable audit log feature distributes multiple dedicated log files, as shown below, and executes multiple output processes (logger processes) in a one-to-one relationship with those files. The number of distributions can be changed by parameter settings.



Information

Because there is a one-to-one relationship between the dedicated log file and the logger process, the logger process also operates as many dedicated log files as there are. A number is appended to the end of the its command title as reported by ps or Process Explorer, as follows. Use this information as a reference when monitoring the process status.

Example

```
pgaudit logger 0
pgaudit logger 1
pgaudit logger 2
```

Parameters in the pgaudit configuration file

Set the following parameters to the "output section":
Refer to the "6.4 pgaudit Configuration File" for details on the parameters.

enable_parallel_logger

Set "on". The default is "off".
If the logger parameter in the pgaudit configuration file is not "auditlog" (the default), it is ignored even if set to "on"

parallel_loggers

Sets the number of logger processes. Specify a value greater than or equal to "2."
Refer to "6.4 pgaudit Configuration File" for an estimate of the values to set.

log_rotation_age

We strongly recommend that you set this parameter. Although the scalable audit log feature works without configuration, it is easier to avoid incorrect analysis when analyzing the output audit logs. Refer to "Considerations when Using the Scalable Audit Log Feature".

Example

```
[output]
enable_parallel_logger = on
parallel_loggers = 5
log_rotation_age = 1h
```

postgresql.conf Parameters

max_worker_processes

Adds the value set for parallel_loggers to the value set for the max_worker_processes parameter. This is because the logger process acts as a background worker.

Example

If max_worker_processes is set to 8 and you want to set parallel_loggers to 3, do the following:

```
max_worker_processes = 11
```

Storage location of the dedicated log file

A dedicated log file is output with the following path and name:
For more information about the log_directory and log_filename parameters, refer to "6.4 pgaudit Configuration File".

```
log_directory parameter setting/number/number-log_filename parameter setting
```

The number is a number between 0 and the number specified by parallel_loggers. If parallel_loggers is 3, it can be 0 or greater and 2 or less. For more information about how to analyze these multiple files, refer to "6.9 Analyzing Audit Logs in SQL".

Example

If you set parallel_loggers=3, a dedicated log file is generated as follows:

```
pgaudit_log/0/0-pgaudit-2024-02-02_153000.log
pgaudit_log/1/1-pgaudit-2024-02-02_153000.log
pgaudit_log/2/2-pgaudit-2024-02-02_153000.log
```

If you can use multiple disks, you can also distribute the disk I/O load by setting the dedicated log file storage directory as a symbolic link to another disk, as shown below.
For Windows, use junction instead of symbolic links.

Example

Place pgaudit_log/2 on a different disk:

L

```
ln -s /other_disk/2  pgaudit_log/2
```

W

```
mklink /J pgaudit_log\2 E:\other_disk\2
```

Depending on the situation, create a symbolic link as follows. If the directory (or symbolic link) specified as the log output destination for the scalable audit log feature does not exist, the directory is automatically created. If the directory already exists, it is used as the log output destination.

- If you are using the scalable audit log feature for the first time, there is no dedicated log file directory. In this case, create a symbolic link after stopping the instance.
- If you are already using the scalable audit log feature, you already have a dedicated log file directory. After stopping the instance, move the stored dedicated log file to another location, and then delete the storage directory. Then, create a symbolic link with the same name as the deleted directory.

For Windows, use junction instead of symbolic links.

6.4 pgaudit Configuration File

In the pgaudit configuration file, specify the information required for pgaudit actions. The pgaudit configuration file comprises three sections: "output section", "option section", and "rule section".

output section

The output section is specified using the format below:

- *paramName* = '*value*'

The valid parameters in the output section are shown in the table below.

L

Linux

Parameter name	Description	Remarks
logger	Dedicated log file (auditlog)/*serverLog* (serverlog) that will be the output destination of the audit log The default is "auditlog" (dedicated log file).	The dedicated log file is output using the same encoding as used for the database.
log_directory	Directory where the audit log is to be created Specify the full path or the relative path from the data storage directory. The default is "pgaudit_log". However, make sure that audit log files are not output under the data storage directory. This is because if you are a backup target, such as pg_basebackup, and recover using that backup data, the audit log from the current time to the backup time will disappear. This is fine if the audit log file is located at the end of the symbolic link. This is because backups such as pg_basebackup do not track such symbolic links. If you specify "on" for the enable_parallel_logger parameter, an audit log is created for "log_directory parameter setting/number". Refer to "6.3 Setting Up the Scalable Audit Log Feature" for more information.	Enabled only if "auditlog" is specified for the logger parameter
log_filename	File name of the audit log Specify a file name that varies according to the time, in the same manner as for log_filename in the postgresql.conf file. The default is "pgaudit-%Y-%m-%d_%H%M%S.log". If you specify "on" for the enable_parallel_logger parameter, the audit log filename is "number-log _filename parameter setting". Refer to "6.3 Setting Up the Scalable Audit Log Feature" for more information.	Enabled only if "auditlog" is specified for the logger parameter
log_file_mode	Specify the permissions of the audit log so that only permitted persons can access it. The parameter value is the numeric mode specified in the format permitted in chmod and umask system calls. The default is "0600". Refer to "log_file_mode" in "6.2 Setup" for information on audit log file permissions.	Enabled only if "auditlog" is specified for the logger parameter
log_rotation_age	Maximum age of the audit log file A new audit log file is generated when the time (minute units) specified here elapses. To disable generation of new log files based on time, specify "0". The valid units are "min" (minutes), "h" (hours), and "d" (days). If the unit is omitted, "min" will be used. The default is "1d" (1 day). We strongly recommend that you set this parameter if you want to take advantage of the scalable audit log feature. This is because all logger processes rotate their dedicated log files at the same time, making it easier to	Enabled only if "auditlog" is specified for the logger parameter

Parameter name	Description	Remarks
	avoid incorrect analysis. For more information, refer to "Considerations when Using the Scalable Audit Log Feature".	
log_rotation_size	Maximum size of the audit log file A new log file will be generated after logs of the size specified here are output to a log file. To disable generation of new log files based on size, specify "0". The valid units are "kB" (kilobytes), "MB" (megabytes), and "GB" (gigabytes). If the unit is omitted, "kB" will be used. The default is "10MB".	Enabled only if "auditlog" is specified for the logger parameter
log_truncate_on_rotation	If rotating audit log files based on time, this parameter is used to specify whether to overwrite (on)/not overwrite (off) existing audit log files of the same name. For example, if "on" is specified, and "pgaudit-%H.log" is specified for log_filename, 24 separate log files will be generated based on time, and those files will be cyclically overwritten. The default is "off". If "off" is specified, the logs will be written to the existing audit log files.	Enabled only if "auditlog" is specified for the logger parameter
fifo_directory	FIFO (named pipe) directory to be used between the daemon process that outputs audit log files and the backend process FIFO named p.PGAUDIT.*nnnn* (*nnnn* is the postmaster PID) are created in the fifo_directories directory. The files cannot be deleted manually. The default is "/tmp". If you specify "on" for the enable_parallel_logger parameter, the name of the FIFO is "p. PGAUDIT. nnnn. number". Refer to "6.3 Setting Up the Scalable Audit Log Feature" for more information about the numbers.	Enabled only if "auditlog" is specified for the logger parameter
enable_parallel_logger (*1)	Specify whether to distribute the audit log output load (on)/not (off). The default is "off". The setting of this parameter is ignored when you specify "serverlog" for the logger parameter, because the serverlog of the database server cannot be distributed.	Enabled only if "auditlog" is specified for the logger parameter
parallel_loggers (*1)	Specifies the multiplicity at which the output load of the audit log is to be distributed. You can specify a number greater than or equal to 1, but to distribute the audit log output load, specify a number greater than or equal to 2. The default is "2". Estimate parallel_loggers as follows: parallel_loggers = number of cores / 2 This estimate assumes the highest load to output audit logs, so setting a value less than the estimated value will not significantly degrade performance in most cases. The disadvantage of increasing this value is that it consumes a little more memory. For more information about estimating memory, refer to "FUJITSU Enterprise	Enabled only if "auditlog" is specified for the logger parameter

Parameter name	Description	Remarks
	Postgres Memory Requirements" in the Installation and Setup Guide for Server. Because the logger process acts as a background worker, add the value set for this parameter to the parameter max_worker_processes in postgresql.conf that sets the maximum number of background workers. [Note] If the max_worker_processes setting is insufficient, the instance cannot be started.	

*1) Enabled by applying the bug fix that includes the PH23619.

Windows

Parameter name	Description	Remarks
logger	Dedicated log file (auditlog)/*serverLog* (serverlog) that will be the output destination of the audit log The default is "auditlog" (dedicated log file).	The dedicated log file is output using the same encoding as used for the database.
log_directory	Directory where the audit log is to be created Specify the full path or the relative path from the data storage directory. The default is "pgaudit_log". However, make sure that audit log files are not output under the data storage directory. This is because if you are a backup target, such as pg_basebackup, and recover using that backup data, the audit log from the current time to the backup time will disappear. This is fine if the audit log file is located at the end of the junction. This is because backups such as pg_basebackup do not track such junction. If you specify "on" for the enable_parallel_logger parameter, an audit log is created for "log_directory parameter setting/number". Refer to "6.3 Setting Up the Scalable Audit Log Feature" for more information.	Enabled only if "auditlog" is specified for the logger parameter
log_filename	File name of the audit log Specify a file name that varies according to the time, in the same manner as for log_filename in the postgresql.conf file. The default is "pgaudit-%Y-%m-%d_%H%M%S.log". If you specify "on" for the enable_parallel_logger parameter, the audit log filename is "number-log _filename parameter setting". Refer to "6.3 Setting Up the Scalable Audit Log Feature" for more information.	Enabled only if "auditlog" is specified for the logger parameter
log_file_mode	This parameter is ignored in Windows.	
log_rotation_age	Maximum age of the audit log file A new audit log file is generated when the time (minute units) specified here elapses. To disable generation of new log files based on time, specify "0".	Enabled only if "auditlog" is specified for the logger parameter

Parameter name	Description	Remarks
	The valid units are "min" (minutes), "h" (hours), and "d" (days). If the unit is omitted, "min" will be used. The default is "1d" (1 day). We strongly recommend that you set this parameter if you want to take advantage of the scalable audit log feature. This is because all logger processes rotate their dedicated log files at the same time, making it easier to avoid incorrect analysis. For more information, refer to "Considerations when Using the Scalable Audit Log Feature".	
log_rotation_size	Maximum size of the audit log file A new log file will be generated after logs of the size specified here are output to a log file. To disable generation of new log files based on size, specify "0". The valid units are "kB" (kilobytes), "MB" (megabytes), and "GB" (gigabytes). If the unit is omitted, "kB" will be used. The default is "10MB".	Enabled only if "auditlog" is specified for the logger parameter
log_truncate_on_rotation	If rotating audit log files based on time, this parameter is used to specify whether to overwrite (on)/not overwrite (off) existing audit log files of the same name. For example, if "on" is specified, and "pgaudit-%H.log" is specified for log_filename, 24 separate log files will be generated based on time, and those files will be cyclically overwritten. The default is "off". If "off" is specified, the logs will be written to the existing audit log files.	Enabled only if "auditlog" is specified for the logger parameter
enable_parallel_logger (*1)	Specify whether to distribute the audit log output load (on)/not (off). The default is "off". The setting of this parameter is ignored when you specify "serverlog" for the logger parameter, because the serverlog of the database server cannot be distributed.	Enabled only if "auditlog" is specified for the logger parameter
parallel_loggers (*1)	Specifies the multiplicity at which the output load of the audit log is to be distributed. You can specify a number greater than or equal to 1, but to distribute the audit log output load, specify a number greater than or equal to 2. The default is "2". Estimate parallel_loggers as follows: parallel_loggers = number of cores / 2 This estimate assumes the highest load to output audit logs, so setting a value less than the estimated value will not significantly degrade performance in most cases. The disadvantage of increasing this value is that it consumes a little more memory. For more information about estimating memory, refer to "FUJITSU Enterprise Postgres Memory Requirements" in the Installation and Setup Guide for Server.	Enabled only if "auditlog" is specified for the logger parameter

Parameter name	Description	Remarks
	Because the logger process acts as a background worker, add the value set for this parameter to the parameter max_worker_processes in postgresql.conf that sets the maximum number of background workers. [Note] If the max_worker_processes setting is insufficient, the instance cannot be started.	

*1) Enabled by applying the bug fix that includes the PH23619.

Information

If the logger parameter is set to "serverlog", audit logs will be output to the server log as log messages, therefore the status information and message severity level according to the log_line_prefix parameter in postgresql.conf will be output to the beginning of the audit log.

If the logger parameter is omitted or set to "auditlog", audit logs will be output to a dedicated log file as dedicated logs, therefore the status information and message severity level according to the log_line_prefix parameter in the postgresql.conf file will not be output.

Refer to "Output format" in "6.5 Session Audit Logging"or "Output format" in "6.6 Object Audit Logging" for information on the output format of audit logs.

L

Point

The pgaudit log_file_mode configuration parameter setting is separate from, and unaffected by, the log_file_mode GUC parameter setting and the -g/-allow-group-access initdb option.

When using a dedicated pgaudit log file, since the pgaudit log_directory location defaults to inside the data storage directory, it is possible for the pgaudit log_file_mode permissions to conflict with the intended file permissions specified by the -g/-allow-group-access initdb option. In this case, the pgaudit log_directory should be specified to be a directory located outside of the data storage directory.

W

Point

If the file permissions for the server log do not satisfy the security requirements of the system, specify a dedicated log file for the output destination of the audit log, and set the access permissions for the output destination directory so that the security requirements are satisfied. At that time, assign read and write permissions to the database administrator.

option section

The option section is specified using the format below:

- *paramName* = '*value*'

The valid parameters in the option section are shown in the table below.

Parameter name	Description	Remarks
role	Name of roles used in Object Audit Logging If specifying a name containing uppercase characters, key words, multibyte characters and commas, enclose the name in double quotation marks.	Parameter used in Object Audit Logging only
log_catalog	Whether to enable (on)/disable (off) log output for pg_catalog If pgAdmin, and so on, will not be retrieving audit logs that access pg_catalog, specify "off". The default is "on" (enabled).	
log_parameter	Whether to enable (on)/disable (off) output of values passed by parameters in SQL execution The default is "off" (disabled).	
log_statement_once	Whether to control (on)/not control (off) output for the second and subsequent SQL statements if the same SQL statement is the log output target The default is "off" (do not control).	
log_level	Log level of audit logs The valid values are "DEBUG5", "DEBUG4", "DEBUG3", "DEBUG2", "DEBUG1", "INFO", "NOTICE", "WARNING", and "LOG". The default is "LOG".	Enabled only if "serverlog" is specified for the logger parameter
audit_log_disconnections	When using Mirroring Controller, specify whether to enable (on) or disable (off) the output of disconnection logs for connections other than those of Mirroring Controller. The default is "off" (disabled). This parameter is valid when log_disconnections in postgresql.conf is off.	Parameter used in Session Audit Logging only

rule section

The rule section is used in Session Audit Logging. Refer to "6.5 Session Audit Logging" for details.

Note

Do not specify the rule section if the role parameter has been specified in the option section. If you specify the rule section, the audit logs of Object Audit Logging and Session Audit Logging will be output intermingled. The CSV format of the role audit log and the CSV format of the rule audit log are different, you will be unable to view in CSV format.

6.5 Session Audit Logging

In Session Audit Logging, specify the rules for filtering logs to be output in the rule section in the pgaudit configuration file.

Rules are specified using the formats below. Multiple values can be specified, using a comma as the delimiter.

- *paramName* = '*value*'
- *paramName* != '*value*'

If [rule] is described on its own in the rule section with no parameters specified, all audit logs of Session Audit Logging will be output.

Example

```
[output]
logger = 'auditlog'
[rule]
```

If [rule] is not described as a section, audit logs of Session Audit Logging will not be output.

Example

```
[output]
logger = 'auditlog'
```

The valid parameters in the rule section are shown in the table below.

Parameter name	Description	Example
timestamp	Timestamp range Refer to "timestamp" for details on how to specify timestamps.	timestamp = '09:00:00 - 10:00:00, 18:00:00 - 18:30:00'
database	Database name To specify a blank value, use a pair of double quotation marks ("") instead of the *value*. When specifying a name containing uppercase characters, key words, multibyte characters and commas, enclose the name in double quotation marks.	database = 'prodcut_db'
audit_role	Role name To specify a blank value, use a pair of double quotation marks ("") instead of the *value*. When specifying a name containing uppercase characters, key words, multibyte characters and commas, enclose the name in double quotation marks.	audit_role = 'appuser1'
class	Operation class Select from the values below. Multiple values can be specified. Refer to "class" for details on the meaning of each class. - BACKUP - CONNECT - DDL - ERROR - FUNCTION - MISC - READ - ROLE - WRITE - SYSTEM	class = 'READ, WRITE'
object_type	Object type	object_type = 'TABLE, INDEX'

Parameter name	Description	Example
	This parameter is enabled when the class parameter is "READ" and "WRITE". Select from the values below. Multiple values can be specified. - TABLE - INDEX - SEQUENCE - TOAST_VALUE - VIEW - MATERIALIZED_VIEW - COMPOSITE_TYPE - FOREIGN_TABLE - FUNCTION	
object_name	Object name This parameter is enabled when the class parameter is "READ" and "WRITE". For objects that can be modified by a schema, such as a table, modify such objects by schema name. When specifying a name containing uppercase characters, key words, multibyte characters and commas, enclose the name in double quotation marks. When specifying the object name "*schemaName*.*tableName*", enclose the entire object name modified by schema name in double quotation marks. To specify a blank value, use a pair of double quotation marks ("") instead of the *value*.	object_name = 'myschema.tbl1' object_name = 'myschema.tbl1, "*mySchema*.TABLE"'
application_name	Application name To specify a blank value, use a pair of double quotation marks ("") instead of the *value*.	application_name = 'myapp'
remote_host	Connection source(client side) host name or IP address If "on" is specified for the log_hostname parameter in the postgresql.conf file, specify a host name. Otherwise, specify the IP address. If using a local host, specify "[local]". To specify a blank value, use a pair of double quotation marks ("") instead of the *value*.	remote_host = 'ap_server'

timestamp

Specify a timestamp range from "*startTime*" to "*endTime*" for the log output target. The timestamp format is 'hh:mm:dd-hh:mm:dd' (hh is expressed in 24-hour notation, and hh, mm, and dd are expressed in two-digit notation).

The start time must be earlier than the end time. If specifying multiple ranges, specify each start and end timestamp using a comma as the delimiter.

End timestamps consider milliseconds. For example, if '11:00:00 - 11:59:59' is specified for the timestamp, "11:00:00:000" to "11:59:59:999" will be the target range.

The timestamps used by evaluation in the rule section of pgaudit are different to the timestamps issued in the log entries. That is because log entries are output after evaluation by pgaudit, with the timestamp being generated at that time.

class

The meaning of each class specified in the class parameter is below:

- READ: SELECT, COPY FROM
- WRITE: INSERT, UPDATE, DELETE, TRUNCATE, COPY TO
- FUNCTION: Function call, DO
- ROLE: GRANT, REVOKE, CREATE ROLE, ALTER ROLE, DROP ROLE
- DDL: All DDLs (such as CREATE and ALTER) other than the DDLs of the ROLE class
- CONNECT: Events relating to connecting (request, authenticate, and disconnect)
- SYSTEM: Instance start, promotion to primary server
- BACKUP: pg_basebackup
- ERROR: Event completed by an error (PostgreSQL error codes other than 00). This class can be used if ERROR or lower level is specified for the log_min_messages parameter in postgresql.conf.
- MISC: Other commands (such as DISCARD, FETCH, CHECKPOINT, and VACUUM)

Evaluation of the rule section

- When a log event occurs, all expressions in the rule section are evaluated at once. Log entries are only output if all parameters in the rule section are evaluated as being true.

  For example, if the rule below has been set, of the operations performed by 'apserver' to 'myschema.tbl1', the operations applicable to classes other than 'WRITE" in the period from 10 a.m. to 11 a.m. will be output as audit logs.

  ```
  [rule]
  timestamp = '10:00:00-11:00:00'
  remote_host = 'apserver'
  object_name = 'myschema.tbl1'
  class != 'WRITE'
  ```

- Multiple rule sections can be defined in the pgaudit configuration file. Log events are evaluated using each rule section, and an audit log is output for each matching rule section.

  For example, if the rules below are set, duplicated audit logs will be output.

  ```
  [rule]
  object_name = 'myschema.tbl1'
  [rule]
  object_name = 'myschema.tbl1'
  ```

- If the same parameter is specified multiple times in one rule section, the last specified parameter is effective.

  For example, if the rule below has been set, "object_name = 'myschema.tbl3'" will take effect.

```
[rule]
object_name = 'myschema.tbl1'
object_name = 'myschema.tbl2'
object_name = 'myschema.tbl3'
```

Output format

In Session Audit Logging, audit logs are output in the format below:

```
AUDIT: SESSION,READ,2022-03-12 19:00:58 PDT,
   (1)         (2)     (3)
[local],19944,psql,appuser,postgres,2/8, 2, 1,SELECT,,TABLE,myschema.account, ,
  (4)    (5)   (6)   (7)      (8)   (9)(10)(11)(12)(13)(14)    (15)        (16)
SELECT * FROM myschema.account;,<not logged>
       (17)                       (18)
```

No	Content
(1)	Log header Fixed as "AUDIT: SESSION".
(2)	Class
(3)	SQL start time
(4)	Remote host name If using a local host, [local] is output.
(5)	Backend process ID
(6)	Application name If an application name is not specified, [unknown] is output.
(7)	User name
(8)	Database name
(9)	Virtual transaction ID
(10)	Statement ID
(11)	Substatement ID
(12)	Command tag
(13)	SQLSTATE
(14)	Object type
(15)	Object name
(16)	Error message
(17)	SQL If the SQL contains a password, such as for CREATE ROLE, and so on, it will be replaced with "<REDACTED>". Additionally, if "on" is specified for the log_statement_once parameter of the option section in the pgaudit configuration file, "<previously logged>" is output for the second and subsequent statements.
(18)	Depending on the log_parameter parameter value of the option section in the pgaudit configuration file, the output content will be as below. - log_parameter=on is specified

No	Content
	If parameters are specified in the SQL, the parameters are concatenated and output, using a space as the delimiter. If parameters are not specified in the SQL, "<none>" is output. - log_parameter=off (default) is specified "<not logged>" is output. Additionally, if "on" is specified for the log_statement_once parameter of the option section in the pgaudit configuration file, "<previously logged>" is output for the second and subsequent statements.

Information

If accessing resources that use the features below, the command tag (12) may be output as "???".

- INSTEAD OF trigger
- RULE
- VIEW
- Security policy per row
- Table inheritance

Example

Below is an example of retrieving audit logs in Session Audit Logging.

1. Settings

   In the pgaudit configuration file, specify the rule section below.

```
[rule]
class = 'READ, WRITE'
object_name = 'myschema.account'
```

2. Retrieving logs

   Execute the SQL below from the client.

```
CREATE TABLE myschema.account
(
    id int,
    name text,
    password text,
    description text
);
INSERT INTO myschema.account (id, name, password, description) VALUES (1, 'user1',
'HASH1', 'blah, blah');
SELECT * FROM myschema.account;
```

   The audit log below can be retrieved.

   'DDL' is not defined in the class parameter, so CREATE TABLE is not output as an audit log.

```
AUDIT: SESSION,WRITE,2022-03-12 19:00:49 PDT,[local],19944,psql,appuser,postgres,
2/7,1,1,INSERT,,TABLE,myschema.account,,"INSERT INTO myschema.account (id, name,
password, description) VALUES (1, 'user1', 'HASH1', 'blah, blah');",<not logged>
```

```
AUDIT: SESSION,READ,2022-03-12 19:00:58 PDT,[local],19944,psql,appuser,postgres,
2/8,2,1,SELECT,,TABLE,myschema.account,,SELECT * FROM myschema.account;,<not logged>
```

6.6 Object Audit Logging

In Object Audit Logging, retrieval of audit logs is achieved by using roles.

Roles are specified in the role parameter of the option section to retrieve audit logs. If there are privileges for commands executed by a role, or if privileges have been inherited from another role, those command operations are output as audit logs.

For example, after "auditor" is set for the role parameter of the option section, the SELECT and DELETE privileges for the account table are assigned to "auditor". In this case, when SELECT or DELETE is executed for the account table, audit logs are output.

Output format

In Object Audit Logging, audit logs are output in the format below:

```
AUDIT: OBJECT,1,1,READ,SELECT,TABLE,public.account,SELECT password FROM account;,<not
logged>
     (1)    (2)(3)(4)  (5)    (6)        (7)              (8)                    (9)
```

No	Content
(1)	Log header Fixed as "AUDIT: OBJECT".
(2)	Statement ID
(3)	Substatement ID
(4)	Class name
(5)	Command tag
(6)	Object type
(7)	Object name
(8)	SQL If "on" is specified for the log_statement_once parameter of the option section in the pgaudit configuration file, "<previously logged>" is output for the second and subsequent statements.
(9)	Depending on the log_parameter parameter value of the option section in the pgaudit configuration file, the output content will be as below. - When log_parameter=on If parameters are specified in the SQL, the parameters are concatenated and output, using a comma as the delimiter. If parameters are not specified in the SQL, "<none>" is output. - When log_parameter=off (default) "<not logged>" is output. Additionally, if "on" is specified for the log_statement_once parameter of the option section in the pgaudit configuration file, "<previously logged>" is output for the second and subsequent statements.

Information

If accessing resources that use the features below, the command tag (5) may be output as "???".

- INSTEAD OF trigger
- RULE
- VIEW
- Security policy per row
- Table inheritance

Example

Below is an example of retrieving logs in Object Audit Logging.

By setting the target for assigning privileges to roles in detail, log output can be controlled.

In the example below, log retrieval of the account table is controlled by the privileges assigned to the columns, however, log retrieval of the account_role_map table is controlled by the privileges assigned to the table.

1. Settings

The role parameter below is specified for the option section in the pgaudit configuration file.

```
[option]
role = 'auditor'
```

2. Defining a role

A role is defined for Object Audit Logging.

```
CREATE USER auditor NOSUPERUSER LOGIN;
```

3. Retrieving logs

Execute the SQL below from the client.

```
CREATE TABLE account
(
    id int,
    name text,
    password text,
    description text
);
GRANT SELECT (password) ON public.account TO auditor;
SELECT id, name FROM account;
SELECT password FROM account;
GRANT UPDATE (name, password) ON public.account TO auditor;
UPDATE account SET description = 'yada, yada';
UPDATE account SET password = 'HASH2';
CREATE TABLE account_role_map
(
    account_id int,
    role_id int
);
GRANT SELECT ON public.account_role_map TO auditor;
SELECT account.password, account_role_map.role_id
  FROM account
  INNER JOIN account_role_map ON account.id = account_role_map.account_id;
```

The audit log below can be retrieved.

In the account table, only the operations for columns that privileges have been assigned to are output as logs.

In the account_role_map table, privileges are assigned to the table, so operations performed for the table are output as logs.

```
AUDIT: OBJECT,4,1,READ,SELECT,TABLE,public.account,SELECT password FROM account;,<not
logged>
AUDIT: OBJECT,7,1,WRITE,UPDATE,TABLE,public.account,UPDATE account SET password =
'HASH2';,<not logged>
AUDIT: OBJECT,10,1,READ,SELECT,TABLE,public.account,"SELECT account.password,
account_role_map.role_id
        FROM account
        INNER JOIN account_role_map ON account.id = account_role_map.account_id;",<not
logged>
AUDIT: OBJECT,10,1,READ,SELECT,TABLE,public.account_role_map,"SELECT account.password,
account_role_map.role_id
        FROM account
        INNER JOIN account_role_map ON account.id = account_role_map.account_id;",<not
logged>
```

6.7 Database Multiplexing

This section describes audit log retrieval while in database multiplexing mode.

6.7.1 Setup

If setting up the audit log feature in a database multiplexing environment that has already been built, follow the procedure below.

1. Copy the pgaudit files

   Copy the pgaudit files on the primary server and standby server.

   Refer to step 1 in "6.2 Setup" for details on copying the pgaudit files.

2. Create the pgaudit configuration file

   Create the pgaudit configuration file on the primary server. Copy the pgaudit configuration file you created to the standby server.

   Refer to step 2 in "6.2 Setup" for details on creating the pgaudit configuration file.

3. Configure postgresql.conf

   In the postgresql.conf file on the primary server and standby server, configure the parameters for using audit logs. Set the same values for the parameters.

   Refer to step 3 in "6.2 Setup" and "6.7.2 Configuring Audit Log Retrieval" for details on the parameters to configure.

4. Configure the *serverIdentifier*.conf file of Mirroring Controller

   In the *serverIdentifier*.conf file on the primary server and standby server, configure the parameters for using audit logs.

   Refer to "6.7.2 Configuring Audit Log Retrieval" for details on the parameters to be set.

5. Start the instance

   Start the instance of the primary server and standby server.

6. Create the pgaudit extension

   Execute CREATE EXTENSION on the primary server to create a pgaudit extension.

   Refer to step 5 in "6.2 Setup" for details on creating pgaudit extensions.

7. Configure the parameters in the pgaudit configuration file

   Add/change the parameters of the pgaudit configuration file on the primary server. Copy the pgaudit configuration file with the added/changed parameters to the standby server.

Refer to "6.4 pgaudit Configuration File" and "6.7.2 Configuring Audit Log Retrieval" for details on the parameters to set.

8. Restart the instance

   Restart the instance of the primary server and standby server.

6.7.2 Configuring Audit Log Retrieval

In database multiplexing mode, Mirroring Controller periodically accesses the database to check the multiplexing status and detect failure. Due to this, audit logs are also periodically retrieved, so log files become used up. Therefore, set the parameters below so that audit logs are not retrieved by Mirroring Controller.

postgresql.conf

log_connections

Omit, or specify "off".

log_disconnections

Omit, or specify "off".

serverIdentifier.conf file of Mirroring Controller

target_db

Specify "template1".

Note

If creating a new database, create it after stopping Mirroring Controller, or specify a name other than "template1" for the template database.

pgaudit configuration file

option section audit_log_disconnections

If you want to output disconnection logs for connections other than Mirroring Controller, specify audit_log_disconnections = "on".

rule section database

Specify database != 'template1'.

6.8 Failover

This section explains audit log retrieval performed by the failover feature integrated with the cluster software.

6.8.1 Configuring Audit Log Retrieval

The failover feature accesses the database periodically to check the instance status. Due to this, audit logs are also periodically retrieved, so log files become depleted. Therefore, set the parameters below so that audit logs are not retrieved.

postgresql.conf

log_connections

Omit, or specify "off".

log_disconnections

Omit, or specify "off".

pgaudit configuration file

rule section database

Specify database != 'template1'.

6.9 Analyzing Audit Logs in SQL

In general, you can quickly and easily analyze the audit log by creating a table for analysis and loading the audit log into that table.
Maybe the analysis table is not updated, so we recommend creating that table as an UNLOGGED table. This is because when loading, the write-ahead log (WAL) is not output and it is not replicated, so it can be loaded faster without interfering with other tasks.

How to Identify the Audit Log Files to Load

To analyze audit logs for a specific time period, load all files that contain audit logs for that time period.
However, do not access dedicated log files with the latest update date. This is because it contains incomplete audit log data that is being written.

Loading Method

The audit log is output in csv format, and can be loaded using the COPY FROM statement or the pgx_loader command.
The following example shows how to load Session Audit Logging output to a dedicated log file.

1. Creating Tables

   Define a table with the columns required to reference the audit log.

```
$ psql
=# CREATE UNLOGGED TABLE auditlog (
header text,
class text,
sql_start_time timestamp with time zone,
remote_host_name text,
backend_process_id integer,
application_name text,
session_user_name text,
database_name text,
virtual_transaction_id text,
statement_id integer,
substatement_id integer,
command_tag text,
sqlstate text,
object_type text,
object_name text,
error_message text,
sql text,
parameter text
);
```

2. Creating an Index

   For example, creating an index on a timestamp is useful for time-series analysis.

```
$ psql
=# CREATE INDEX auditlog_time_index ON auditlog (sql_start_time);
```

3. Loading the Audit Log

   Specify a dedicated log file in the FROM clause of COPY FROM.

```
$ psql
=# COPY auditlog FROM
'pgaudit-2024-02-02_150000.log' WITH CSV DELIMITER ',';
```

   You can also use the PROGRAM clause to load multiple dedicated log files together.
   The following example loads one day of audit logs by specifying wildcard for day part of file name:

L

```
$ psql
=# COPY auditlog FROM
PROGRAM 'cat pgaudit-2024-02-02_*' WITH CSV DELIMITER ',';
```

W

```
$ psql
=# COPY auditlog FROM
PROGRAM 'type pgaudit-2024-02-02_*' WITH CSV DELIMITER ',';
```

4. Audit Log Analysis

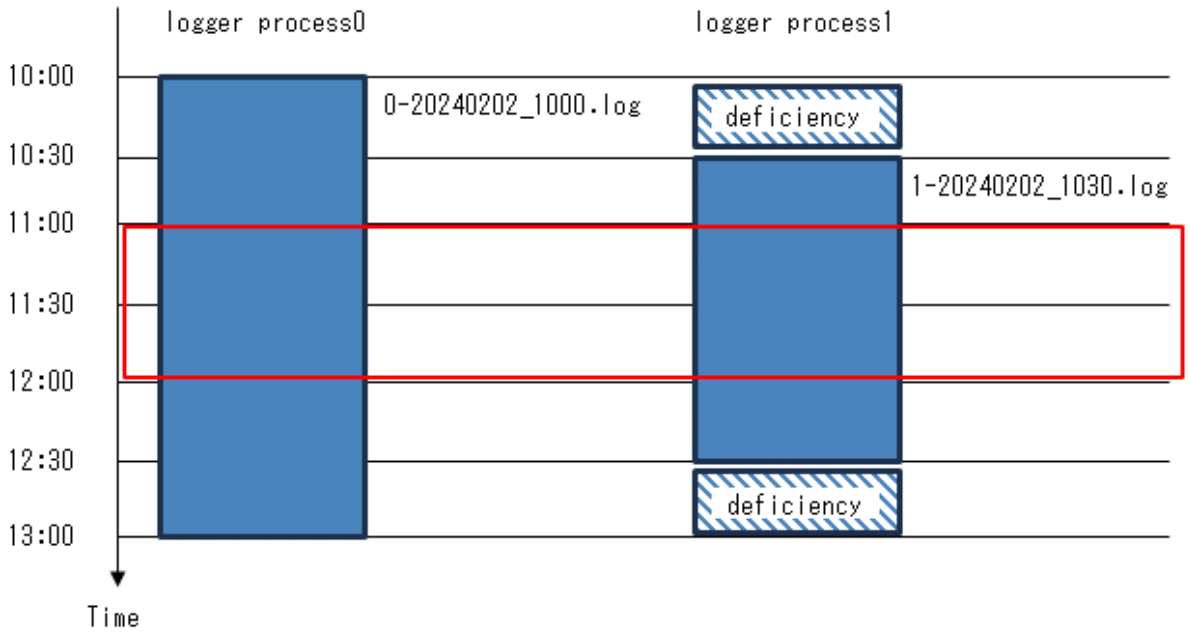
   Access the table that loaded the audit log.
   Loaded records are not in chronological order, but can be sorted by timestamp

```
$ psql
=# SELECT * FROM auditlog ORDER BY sql_start_time;
 header         |  class |        sql_start_time     | remote_host_name  |  ...
----------------+--------+---------------------------+-------------------+  ...
AUDIT: SESSION | DDL    | 2024-02-02 15:00:49+09    | ::1               |  ...
AUDIT: SESSION | SYSTEM | 2024-02-02 15:00:58+09    |                   |  ...
```

Considerations when Using the Scalable Audit Log Feature

- When you enable the scalable audit log feature, there are many dedicated log files for the target period as you specify for parallel_loggers, so make sure to load all the dedicated log files.

- When you disable log_rotation_age, do not analyze the time except target period because not all audit logs may have been loaded.
  For example, suppose that you are analyzing the audit log at 11:00-12:00, and you load the file output by logger process 0 at 10:00-13:00 and the file output by logger process 1 at 10:30-12:30, as shown in the following figure. Audit log records up to 10:00-13:00 appears in the table loaded with log files, but some audit logs (Audit log output by logger process 1 at 10:00-10:30 and 12:30-13:00) have not been loaded.
  When log_rotation_age is enabled, the timing of the dedicated log file changeover is consistent across all logger processes to prevent such erroneous analysis.



6.10 Removing Setup

This section describes how to remove the setup of pgaudit.

1. Start the instance
2. Remove the pgaudit extension

   Execute DROP EXTENSION to remove the pgaudit extension from the database.

```
$ psql -d <database name>
=# DROP EXTENSION pgaudit;
=# \q
```

3. Change postgresql.conf

   Remove the parameters below relating to pgaudit.

   - shared_preload_libraries
   - pgaudit.config_file

4. Restart the instance
5. Remove the pgaudit files

L

As superuser, run the following command. Note that "<*x*>" in paths indicates the product version.

```
$ su -
Password:******
# rm -rf /opt/fsepv<x>server64/filesCopiedDuringSetup
```

Information

The files copied during setup can be checked below.

```
# find /opt/fsepv<x>server64/OSS/pgaudit
```

W

Open a command prompt as administrator privileges, run the following command. Note that "<*x*>" in paths indicates the product version.

```
> del "c:\Program Files\Fujitsu\fsepv<x>server64\filesCopiedDuringSetup"
```

Information

The files copied during setup can be checked below.

```
> dir /b /s "c:\Program Files\Fujitsu\fsepv<x>server64\OSS\pgaudit"
```

Chapter 7 Confidentiality Management

In order to prevent unauthorized use of the various data stored in the database, it is necessary to set appropriate privileges for each database resource for database users. However, there are multiple users to whom privileges are granted, and there are also many target database resources. So it takes a lot of effort to set it up. The confidentiality management feature reduces the time and effort required to do so and supports the setting and maintenance of appropriate privileges.

This chapter describes how to install, design and use confidentiality management features. For usage examples of the feature, refer to "7.8 Usage Example of Confidentiality Management".

7.1 Setup

This feature is provided as an EXTENSION of PostgreSQL. The name is pgx_confidential_management_support. Register the extension with the database cluster using the CREATE EXTENSION statement as follows:

This must be run by superuser. Because this extension registers PostgreSQL event triggers. By registering an event trigger, when a database object such as a table is deleted, related information is deleted from the information managed by the confidentiality management feature.

This extension can be created for any schema.

```
CREATE EXTENSION pgx_confidential_management_support
```

Note

- The various definitions described below are registered in the tables included in this extension. Note that dropping this extension with the DROP EXTENSION statement will therefore drop all these definitions as well.
- If the superuser also serves as the confidentiality management role, or if you have only one confidentiality management role, setup is complete. For confidentiality management roles, refer to "7.2.2 Determining Confidentiality Management Roles". However, if multiple confidentiality management roles manage different confidentiality matrices, they must be prevented from manipulating each other's confidentiality matrices. To do so, a superuser should run the script provided by the product as follows. This script defines policies for the row level security feature on the tables provided by the confidentiality management. ${install_dir} refers to the directory where you installed the product.

  ```
  psql -f ${install_dir}/share/extension/pgx_confidential_management_support_policy.sql
  ```

- 'public' is granted SELECT on tables included by this extension when CREATE EXETENSION statement is executed. Don't revoke the privilege from 'public'. For example, event trigger of the extension confirms to update its tables when a general user drop some table. The privilege is required at the time. Also, there is no problem that users refer content of the tables like pg_catalog.

7.2 Designing Confidentiality Management

Describes the design of confidentiality management.

7.2.1 Designing a Confidentiality Matrix

A confidentiality matrix is a matrix of confidentiality levels and confidentiality groups. Elements of the confidentiality matrix are confidentiality privileges. Here we define them.

A confidentiality level is a group of data with the same degree of confidentiality, and a confidentiality group is a group of roles with the same access to confidentiality data. The details are described below.

The end result is to classify database objects into confidentiality levels and roles into confidentiality groups. But for the first step, define your confidentiality matrix abstractly without thinking about concrete tables, roles, etc. By doing so, the relationship between confidentiality levels and confidentiality groups can also be applied to databases with different sets of database objects.

You can define multiple confidentiality matrices. These confidentiality matrices are identified by a name, but the name is unique within the database (not the database cluster).

A role that manages the confidentiality matrix is called a confidentiality management role. The details are described below.

How to create multiple confidentiality matrices

There are various design methods, but an example is shown below.

- Create one confidentiality matrix for a dataset

  For example, create one confidentiality matrix for one section of your organization.

- Copy the confidentiality matrix if the two sections have the same confidentiality design
- When multiple organizations share datasets, create a single confidentiality matrix for the shared datasets

  An example of this case is shown below.

  - Create a confidentiality matrix *A* for the dataset that only section 1 has access to.
  - Create a confidentiality matrix *B* for the dataset that only section 2 has access to.
  - Create a confidentiality matrix *S* for the dataset shared by section 1 and section 2.
  - Section 1 administrator *M1* is in charge of the confidentiality management role for confidentiality matrix *A*.
  - Section 2 administrator *M2* is in charge of the confidentiality management role for confidentiality matrix *B*.
  - Role group *G* to which *M1* and *M2* belong is in charge of confidentiality management role of confidentiality matrix *S*.

7.2.1.1 Defining Confidentiality Levels

Confidentiality level is a concept that indicates degree of confidentiality. For example, determine a confidentiality level that classifies tables that contain personal information and those that do not.

A collection of data that you classify as confidentiality level is a database object, such as a table. Such objects are called confidentiality objects. Refer to "What is a confidentiality objects?" for confidentiality objects.

Attributes can be set for the confidentiality level. For example, if a confidentiality level has an attribute that requires encryption, tables belonging to that confidentiality level are required to be encrypted. The confidentiality management feature may or may not automatically change the attribute of the confidentiality object to which it belongs. Also, each attribute targets a different confidentiality object.

The table below shows the attribute description, the confidentiality object targeted, and whether to change the attribute of the confidentiality object automatically.

Confidentiality level attributes	Specifiable values	Confidentiality object type	Auto change	Description
encryption_algorithm	AES128 AES256 none	table column rowset	No	Requires confidentiality objects to be encrypted with the specified algorithm. For details, refer to "Protecting Storage Data Using Transparent Data Encryption" in the Operation Guide. If ***none*** is specified, no encryption is requested. Default is ***none***. The attribute is not changed autlmatically because encryption and decryption involve large data updates.

Confidentiality level attributes	Specifiable values	Confidentiality object type	Auto change	Description
				Instead, you cannot add a confidentiality object with an encryption strength lower than the specified encryption strength. Also, when changing to increase the encryption strength, confidentiality objects that do not meet the post-change conditions must not be included.

What is a confidentiality objects?

Confidentiality objects have the types shown in the following table. For example, some data contained in table *T* can be added to confidentiality level *L1* and other data to confidentiality level *L2*.

Confidentiality object type	Description
schema	It does not mean all the tables contained in the schema. It simply means an object that exists in the system catalogs. Means the schema itself that is the target of access control by the GRANT statement.
table	Includes views, materialized views and partitions as well as tables. Currently do not support foreign tables.
column	Means column.
rowset	A set of rows that satisfy a specified condition. Although there is no rowset object in the PostgreSQL system catalog, it is treated as a confidentiality object in the confidentiality management feature for ease of use. Access control for rowset-type confidentiality objects uses PostgreSQL's row-level security functionality internally. Therefore, how access is controlled follows the row-level security specification.

The following objects may also contain confidential information. However, the current confidentiality management feature does not support them as confidentiality objects. When managing these, it is recommended to manage them in a role different from the confidentiality management role. The details are described in "7.5 Suggestions for Monitoring Methods".

- Function
- Procedure
- Foreign server
- Foreign data wrapper
- Foreign table

7.2.1.2 Defining Confidentiality Groups

A confidentiality group is an object that indicates which confidentiality level roles in confidentiality group can acdess to. For example, decide which role groups have access to personal information and which groups do not. Confidentiality groups are actually automatically defined as PostgreSQL roles (role groups). This role is called a confidentiality group role.

What is a confidentiality group role?

Confidentiality group roles are targets of GRANT and REVOKE statements executed internally by the confidentiality management feature. And when you add a role to the confidentiality group, the added role becomes a member of the

confidentiality group role. The privileges of the added role are given by inheriting the privileges given to the confidentiality group role by the confidentiality management feature.

The following attributes can be set for the confidentiality group role via the function that creates the confidentiality group. The meaning and default value of attribute are the same as in the CREATE ROLE statement specification. These are limited to attributes that may be necessary when managing confidentiality. Therefore, use the confidentiality management feature to change it. The confidentiality management feature automatically sets the following attributes for the role added to the confidentiality group.

- SUPERUSER
- CREATEDB
- CREATEROLE
- REPLICATION
- BYPASSRLS

Other attributes follow the PostgreSQL CREATE ROLE statement defaults. Even if the attribute is changed using the ALTER ROLE statement, the operation of confidentiality management feature will not be disturbed.

The naming convention for confidentiality group roles is as follows.

```
pgx_cgroup_role_${serial_number}
```

When deleting a confidentiality group, you can choose not to delete the confidentiality group role. In such a case, it is a means to know that it was a confidentiality group role later.

Note

Do not define roles using strings that follow this naming convention. This is because the name of the new confidentiality group role will be the number next to the maximum *${serial_number}* of the role that follows this naming convention. For example, if a role named 'pgx_cgroup_role_999999999999999999' exists, you will not be able to create new confidentiality group roles.

7.2.1.3 Defining Confidentiality Privilege

Confidentiality privileges indicate how a confidentiality group can access confidentiality objects registered at a certain confidentiality level.

For example, a confidentiality group *R* can be defined to have SELECT, INSERT, UPDATE, and DELETE privileges on table-type confidentiality objects belonging to confidentiality level *L*.

Once a confidentiality object or role has been added to a confidentiality level or confidentiality group, the feature automatically uses the GRANT or CREATE POLICY statement to grant privileges to the actual database object according to the confidentiality privilege definition to the confidentiality group role.

For rowset types, confidentiality privileges are those that can be specified in the CREATE POLICY statement. In cases other than rowset, it is a privilege that can be specified in the GRANT statement. For detailed privileges, refer to the respective SQL statement reference in the PostgreSQL Documentation.

Note

Do not change the definition of POLICY that this feature creates internally. Even if you change it, the function of this feature may return to the state before the change without warning.

Point

For roles registered in the confidentiality matrix, if the privilege to a certain confidentiality object is more than the confidentiality privilege, revoke those privileges according to the confidentiality privilege.

If a role not registered in the confidentiality matrix or PUBLIC was granted privileges to a confidentiality object registered in the confidentiality matrix, functions of this feature for adding confidentiality objects or roles will fail. Because this function is not allowed to manage.

Even in a slightly more complicated situation, failing functions that add confidentiality objects and roles prevent roles added to the confidentiality matrix from deviating from confidentiality privileges:

- It is set so that a role registered in the confidentialmatrix can inherit a role not registered in the confidentiality matrix, and as a result, it has more privileges than the confidentiality privileges.
- Roles registered in the confidentiality matrix are granted privileges by a POLICY not created by this feature. In this case, the function will fail if such a POLICY exists without rigorously checking for violations of confidentiality privilege.

7.2.2 Determining Confidentiality Management Roles

The confidentiality management role performs all operations on the confidentiality matrix. As such, the confidentiality management role requires very strong privileges to either:

1. Has all of the following rights
   - Superuser privileges
   - Ownership of database objects belonging to the confidentiality level
2. Has all of the following privileges
   - CREATEROLE privilege
   - SELECT, INSERT, UPDATE, and DELETE privileges on all tables included in the extension 'public' is granted SELECT when CREATE EXTENSION statement is executed.
   - Ownership of database objects belonging to the confidentiality level

Note

Note that the previous owner may not be able to execute GRANT statements, etc. once the confidentiality management role becomes the owner. This is because ownership of database objects cannot be shared by multiple roles belonging to different role groups. This is the PostgreSQL specification.

The confidentiality management role can create and manage multiple confidentiality matrices, but it is safer to distribute the authority. For that reason, we recommend that you decide on confidentiality management rules in the following order of priority.

- Create one-to-one confidentiality matrices and confidentiality management roles.
- Create a role group with multiple confidentiality management roles as members, or manage multiple confidentiality matrices with one confidentiality management role.
- A superuser is also a confidentiality management role.

Note

To assign attributes that only a superuser can grant to a role managed using the confidentiality management feature, the superuser must also serve as the confidentiality management role. For example, the REPLICATION attribute is such an attribute. Refer to the reference of the PostgreSQL Documentation for details.

7.2.3 Classify Confidentiality Objects According to the Definition of Confidentiality Level

7.2.3.1 Defining Confidentiality Objects

First, refer to list of definitions of schemata, tables in your database, etc. that you want to manage, and define confidentiality objects. As mentioned earlier, there are various types of confidentiality objects such as column type and rowset type, so you can define confidentiality objects based on their data content.

Access control for rowset-type confidentiality objects uses PostgreSQL's row-level security functionality internally. Therefore, the method of specifying a set of rows follows the specifications of the AS clause, USING clause, and WITH CHECK clause of the CREATE POLICY statement. Also, the definition of the rowset-type confidentiality object is specified in the argument of the function that registers the rowset-type confidentiality object to the confidentiality level. This feature executes the ALTER TABLE statement with the ENABLE ROW LEVEL SECURITY clause to enable POLICY when a rowset-type confidentiality object is added.

Currently it is not possible to register a foreign table as a table-type confidentiality object.

Point

Recommend that you do not grant privileges on confidentiality objects to PUBLIC. Granting privileges to PUBLIC is the same as granting privileges to all roles registered in the confidentiality matrix. This makes no sense if all roles that access confidentiality objects are managed using this feature. If PUBLIC is granted privileges to confidentiality objects, the functions included in this feature will check that the privileges granted to each role are not exceeded, and will fail if they are.

Note

- Be careful when confidentiality objects are of column type. This is because if the table-type confidentiality object is set at the same time, the table type privilege takes precedence.

  For example, if you want to revoke the SELECT privilege only from a special column *C* in some table *T*, list the columns other than column *C* and grant the SELECT privilege to them without granting the SELECT privilege to table *T*. This follows the PostgreSQL's GRANT statement specification for column.

  If you have to enumerate a large number of columns, it might be a good idea to move the special columns to a new table and present your existing application with a VIEW that JOINs both tables.

- If the confidentiality object is of rowset type, set the same privileges for the table type as those specified for the rowset type. This is because when a SQL statement accesses data, it first checks that you have privilege to access the table. After that, the rowset privileges are checked for matching rows. This follows the PostgreSQL row level security specification.

7.2.3.2 Classify Confidentiality Objects

Classify confidentiality objects according to the confidentiality level definition.

A single confidentiality object cannot be classified in multiple confidentiality levels, even if they are in different confidentiality matrices.

7.2.4 Classify Roles According to Confidentiality Group Definitions

A role can belong to multiple confidentiality groups in different confidentiality matrices at the same time. However, a single role cannot be classified into multiple confidentiality groups within a single confidentiality matrix.

Note

- We strongly recommend that you do not group confidentiality group roles into other confidentiality groups. This is because the confidentiality management feature does not prohibit such usage, but it probably only complicates security management. Also note that PostgreSQL does not allow cyclic groupings of roles.
- After adding roles to a confidentiality group, it is highly recommended that such roles not be made members of role groups not managed by the confidentiality matrix. This is because although the confidentiality management feature does not prohibit such situations, for example, unreasonably increasing the privileges of such role groups may become a way out for roles managed by the confidentiality management feature.
  Note that this feature does not allow roles with such loopholes to be added to confidentiality groups.

7.3 How to Use Confidentiality Management Feature (Definition)

Confidentiality management feature provides functions and tables.

Some functions define, change, or drop confidentiality matrices, etc. These functions will output an audit log indicating that they were executed, so you can later confirm that an illegal operation was performed.

In addition, you can directly refer to the table provided by this feature and check the defined contents using functions that help referencing.

Also, some functions output the attributes of confidentiality objects defined in the PostgreSQL system catalog and the attributes that should be set for that confidentiality object. You can use these to compare attributes.

For details, refer to "Appendix B System Management Functions Used by Confidentiality Management Feature".

Tables included by the pgx_confidential_management_support extension

Table name	Description
pgx_confidential_matrix	A list of confidentiality matrices. You can refer to the attributes of the registered confidentiality matrix, the update time, or the time when the confidentiality level or confidentiality group was registered or deleted.
pgx_confidential_level	A list of confidentiality levels. You can refer to the registered confidentiality level attributes, update time, or the time when a confidentiality object was registered to the confidentiality level or removed from the confidentiality level.
pgx_confidential_group	A list of confidentiality groups. You can refer to the registered confidentiality group attributes, update time, or the time when a role was registered to the confidentiality group or removed from the confidentiality group.
pgx_confidential_privilege	A list of confidentiality privileges. You can refer to confidentiality privilege set for each confidentiality object, update time, and so on.
pgx_confidential_object	A list of confidentiality objects. You can refer to object attributes or update time, and so on.
pgx_confidential_role	A list of roles registered in the confidentiality group. You can refer to role attributes or update time, and so on.

Table name	Description
pgx_confidential_policy	This is a list of policies created to set privileges for rowset-type confidentiality objects. You can refer to the name of the policy you created and the privileges it has set. Rows in this table are inserted when you add a rowset-type confidentiality object.

Rows in each table are added, deleted, or updated when you execute functions to add, delete, or update definitions.

Also, if the target confidentiality object is deleted by a DROP TABLE statement, etc., it will also be deleted from the following tables.

- pgx_confidential_object
- pgx_confidential_policy

Functions for adding, removing or updating definitions

These functions will print an audit log indicating that they were executed.

Function name	Description
pgx_create_confidential_matrix	Create a confidentiality matrix.
pgx_alter_confidential_matrix	Change the attributes of the confidentiality matrix.
pgx_drop_confidential_matrix	Drop the confidentiality matrix.
pgx_copy_confidential_matrix	Copy the confidentiality matrix.
pgx_create_confidential_level	Create confidentiality levels and register them in the confidentiality matrix.
pgx_alter_confidential_level	Change the confidentiality level attribute.
pgx_drop_confidential_level	Remove a confidentiality level from the confidentiality matrix and drop a confidentiality level.
pgx_create_confidential_group	Create a confidentiality group and register it in the confidentiality matrix.
pgx_alter_confidential_group	Change the attributes of a confidentiality group.
pgx_drop_confidential_group	Remove confidentiality groups from the confidentiality matrix and drop confidentiality group.
pgx_grant_confidential_privilege	Grant confidentiality privileges.
pgx_revoke_confidential_privilege	Revoke confidentiality privileges.
pgx_add_object_to_confidential_level	Add confidentiality objects to the confidentiality level.
pgx_remove_object_from_confidential_level	Removes confidentiality objects from the confidentiality level.
pgx_add_role_to_confidential_group	Add a role to a confidentiality group.
pgx_remove_role_from_confidential_group	Remove a role from a confidentiality group.

Functions that Support Definition Referencing and Comparison with System Catalogs

Function name	Description
pgx_get_attribute_of_objects	For all confidentiality objects registered in the specified confidentiality matrix, displays the

Function name	Description
	attributes defined in the confidentiality matrix and the attributes actually set in the database.
pgx_get_attribute_of_roles	For all roles registered in the specified confidentiality matrix, displays the attributes defined in the confidentiality matrix and the attributes actually set in the system catalog.
pgx_get_privileges_on_level_and_group	Displays a list of combinations of confidentiality objects registered with the specified confidentiality level and roles registered with the specified confidentiality group, such that the following can be compared. - Privileges specified by the confidentiality privilege settings that should be granted to the specified role - Privileges granted in the actual system catalog
pgx_get_privileges_on_object	For the specified confidentiality object, display a list that allows you to compare the following: - Privileges dictated by confidentiality privilege settings that should be granted to all roles - Privileges granted in the actual system catalog
pgx_get_privileges_on_role	For all confidentiality objects, display a list that allows you to compare: - Privileges specified by the confidentiality privilege settings that should be granted to the specified role - Privileges granted in the actual system catalog
pgx_get_privileges_on_matrix	For all objects registered in the specified confidentiality matrix, display a list that allows you to compare: - Privileges defined by confidentiality privilege settings that should be granted to all roles registered in the confidentiality matrix. - Privileges granted in the actual system catalog

Describe the definition procedure.

7.3.1 Creating a Confidentiality Management Role

Create a confidentiality management role using the CREATE ROLE statement, or use an existing role as a confidentiality management role. Set the privileges and attributes shown in "7.2.2 Determining Confidentiality Management Roles" for the confidentiality management role.

- Use caution when renaming a confidentiality management role. If you want to do so, delete all confidentiality matrices managed by that confidentiality management role, rename the confidentiality management role, and then define the same confidentiality matrix again. Otherwise, you will not be able to operate the confidentiality matrix. For example, you cannot change confidentiality privileges or remove confidentiality objects from the confidentiality level. If you

accidentally renamed it first, change it back to the original name and then proceed as described above. In the future, we will simply allow the changed name of the confidentiality management role to be set in the confidentiality matrix.

- If you want to delete the confidentiality management role, delete the confidentiality management role after deleting the confidentiality matrix. Otherwise, you will not be able to create a confidentiality matrix with the same name because you will not be able to delete the confidentiality matrix left behind. If you accidentally delete the confidentiality management role before deleting the confidentiality matrix, create a confidentiality management role with the same name again and delete the confidentiality matrix, or delete the confidentiality matrix with a role that has SUPERUSER privileges.

7.3.2 Creating a Confidentiality Matrix

Create a confidentiality matrix with comments describing the confidentiality matrix as follows. This comment is stored in the pgx_confidential_matrix table.

```
select pgx_create_confidential_matrix('matrix_foo', 'This matrix is defined for foo.')
```

Point

The role executing this function is considered the confidentiality management role. If this function is executed after SET ROLE, the role specified in SET ROLE will be regarded as the confidentiality management role, not the role that executed SET ROLE.

You can also create a new confidentiality matrix 'matrix_dest' by duplicating the already created confidentiality matrix 'matrix_src' as follows. If you copy the confidentiality matrix and create it, the comments will also be copied. Comments can be changed using the pgx_alter_confidential_matrix function.

```
select pgx_copy_confidential_matrix('matrix_dest', 'matrix_src')
```

You can also check the created matrix by referring to the pgx_confidential_matrix table.

Point

The source must be within the same database. If you want to copy from different databases or different database instances, choose one of the following methods. If you choose the method using COPY statement, confirm the cautions shown below.

- Define using a function in the same way as the original.
- Perform the following steps

  Execute steps other than step 6 with a role that has SUPERUSER privileges.

  1. Set up extensions in the target database.
  2. From the table below, specify the ID of the source confidentiality matrix in the WHERE clause and extract the data using the COPY TO statement.
     - pgx_confidential_matrix
     - pgx_confidential_level
     - pgx_confidential_group
     - pgx_confidential_privilege
  3. Refer to the *pg_sequences* system view and save the value of the *last_value* column of the following SEQUENCE in a file.
     - pgx_confidential_matrix_cmatid_seq
     - pgx_confidential_level_clevid_seq
     - pgx_confidential_group_cgroid_seq
     - pgx_confidential_privilege_cpriid_seq

4. Load the data extracted in step 2 into the above table in the target database using the COPY FROM statement. This operation can be performed by any confidentiality management role in any confidentiality matrix.

5. Use the *pg_catalog.setval* function to set the *last_value* for each SEQUENCE on the copy destination, specifying the value saved in step 3. An example is shown below.

```
SELECT pg_catalog.setval('pgx_confidential_matrix_cmatid_seq', 5)
```

6. Create a new confidentiality matrix using the *pgx_copy_confidential_matrix* function and specifying the confidentiality matrix loaded in step 4 as the copy source. Perform this operation in the confidentiality management role of the newly created confidentiality matrix.

7. Use the *pgx_drop_confidential_matrix* function specifying ***false*** for *drop_role* to drop the confidentiality matrix loaded in step 4.

- If you specify true for *drop_role*, you will not be able to continue to be managed by the original confidentiality matrix. The reason is as follows.

  After completing step 6, you should have:

  a) Confidentiality matrix of source database

  b) the confidentiality matrix loaded in step 4

  c) Confidentiality matrix duplicated in step 6

  The b) and c) confidentiality matrices use different confidentiality group roles. This is the effect of the *pgx_copy_confidential_matrix* function. However, the confidentiality matrices of a) and b) share the same confidentiality group role. Although this is temporary, it is a bad situation. To do so, remove the confidentiality matrix in b) in step 7. If *true* is specified for *drop_role* at this time, the confidentiality group role used in a) will be deleted. Therefore, management by the confidentiality matrix of a) cannot be continued.

- The destination must have just installed this extension with CREATE EXTENSION. This is because the table data contained in this extension contains an ID representing the confidentiality matrix and confidentiality level, and this ID is generated by the SEQUENCE included in the extension. For example, if you have created at least one confidentiality matrix, the confidentiality matrix is numbered with an ID of 1, but that matrix is not the object numbered with an ID of 1 in the copy source.

7.3.3 Adding Confidentiality Levels to the Confidentiality Matrix

Create a confidentiality level and add it to the confidentiality matrix as follows: In the example below, the JSON format third argument requests that confidentiality objects belonging to 'level1' are encrypted. Specify *NULL* if you don't want anything. Refer to "B.2 Confidentiality Level Manipulation Functions" for the content of the request.

```
select pgx_create_confidential_level('matrix_foo', 'level1',
'{"encryption_algorithm":"AES256"}', 'The strongest encryption is required for this
level.')
```

You can also check the added confidentiality level by referring to the pgx_confidential_level table.

7.3.4 Adding Confidentiality Groups to the Confidentiality Matrix

Create a confidentiality group and add it to the confidentiality matrix as follows. Attributes are given to the confidentiality group by the third argument in JSON format. Refer to "B.3 Confidentiality Group Manipulation Functions" for the attributes that can be assigned.

In the example below, a role belonging to this confidentiality group grants the CREATEDB privilege.

```
select pgx_create_confidential_group('matrix_foo', 'group1', '{"CREATEDB":true}', 'Roles
belonging to this confidentiality role are permitted to create db.')
```

You can also check the added confidentiality group by referring to the pgx_confidential_group table.

7.3.5 Granting Confidentiality Privileges to Confidentiality Groups

In the example below, roles belonging to 'group1' are granted SELECT, INSERT and DELETE privileges for tables belonging to 'level1'. Specify privileges for each type of confidentiality object by the fourth argument in JSON format. Refer to "B.4 Confidentiality Privilege Manipulation Functions" for the privileges that can be granted.

```
select pgx_grant_confidential_privilege('matrix_foo', 'level1', 'group1',  '{"schema":
["ALL"], "table":["SELECT","INSERT","DELETE"]}')
```

If a confidentiality object of the target type is registered in the confidentiality level, when this function is executed, the confidentiality management feature internally uses the GRANT statement to grant a confidentiality group role access to confidentiality objects that are registered at a confidentiality level. The effect of the GRANT statement can be checked as follows.

```
select pgx_get_privileges_on_level_and_group('matrix_foo', 'level1', '["role1","role2"]')
```

For the format of the table returned by a function that outputs authority information, such as this function, refer to "B.7 Functions that Support Definition Referencing and Comparison with System Catalogs".

7.3.6 Adding Confidentiality Objects to Confidentiality Level

Classify the intended database object into the appropriate confidentiality level according to the confidentiality level design. In the example below, 'table1' of 'schema1' and 'table2' of 'schema1' are added to 'level1' at the same time. Specify the confidentiality object type and confidentiality object name in the third argument in JSON format. If the type is same, you can enumerate multiple confidentiality objects. You can enumerate multiple objects you want to register for each sensitive object type. In this example, only the *table* type is registered, but *schema* type can also be registered at the same time.

```
select pgx_add_object_to_confidential_level ('matrix_foo', 'level1',
'[{
    "type":"table",
    "object":[
        {
          "schema":"schema1",
          "table":["table1","table2"]
        }
     ]
}]')
```

When specifying a rowset, you must declare what kind of set of rows it is in the value of the *rowset_expression* key. For details on how to specify it, refer to "B.5 Confidentiality Object Manipulation Functions".

When this function is executed, the confidentiality management feature internally uses the GRANT statement or CREATE POLICY statement to grant privileges to the specified confidentiality objects to all confidentiality groups set in the confidentiality matrix. The effect of the GRANT statement can be checked as follows. This example checks the privileges granted on 'table1' in 'schema1' and 'table2' in 'schema1'.

```
select pgx_get_privileges_on_object('matrix_foo',
'[{
    "type":"table",
    "object":[
      "schema":"schema1",
      "table":["table1","table2"]
    ]
}]')
```

7.3.7 Adding Roles to Confidentiality Groups

Classify the intended role into the appropriate confidentiality group according to the confidentiality group design. You can add multiple roles at once. In the example below, role role1 and role role2 are added to confidentiality group 'group1'.

```
select pgx_add_role_to_confidential_group('matrix_foo', 'group1', '["role1","role2"]')
```

The confidentiality management feature adds the specified role to the members of the confidentiality group role.

The feature does not execute GRANT statements for confidentiality objects at this time. This is because we have already executed the GRANT statement between the confidentiality group role and the confidentiality object, and we will not execute the GRANT statement between the individual roles and the confidentiality object.

By using the function below, you can check the privileges that can be exercised by the specified role after inheriting the privilege of the confidentiality group role with the INHERIT attribute on the role side or changing to the confidentiality group role with the SET ROLE statement. In other words, you can check the access privileges that the actually registered roles can exercise. For details on the output format, refer to "B.7 Functions that Support Definition Referencing and Comparison with System Catalogs". The example below checks the privileges of role 'role1' and role 'role2'.

```
select pgx_get_privileges_on_role('matrix_foo',  '["role1","role2"]')
```

7.4 How to Use Confidentiality Management Feature (Change and Deletion)

7.4.1 Renaming Confidentiality Objects

The current confidentiality management feature cannot follow when confidentiality objects are renamed. Therefore, if the name is changed, remove the old name confidentiality object with the *pgx_remove_object_from_confidential_level* function specifying the old name, and register the confidentiality object with the new name using the *pgx_add_object_to_confidential_level* function specifying the new name.

7.4.2 Renaming Roles

The current confidentiality management feature cannot follow when roles are renamed. Therefore, when renaming, exclude the role with the old name with the *pgx_remove_role_from_confidential_group* function specifying the old name, and after renaming the role, register the role with the new name with the *pgx_add_role_to_confidential_group* function specifying the new name.

7.4.3 Deleting Roles

The current confidentiality management feature cannot follow when roles are dropped. Therefore, when dropping, exclude the role with *pgx_remove_role_from_confidential_group* function.

7.4.4 Changing Confidentiality Matrix

Change the name or attributes of the confidentiality matrix as follows. Specify the name and attribute value to be changed in JSON format. You can change the name and multiple attributes at once. Attributes not specified here are not changed.

```
select pgx_alter_confidential_matrix('matrix_foo', '{"name": "matrix_bar",
"comment":"This matrix is defined for bar."}')
```

7.4.5 Deleting Confidentiality Matrix

Delete the confidentiality matrix as follows.

```
select pgx_drop_confidential_matrix('matrix_foo', false, false)
```

By specifying a second or third argument, you can choose to delete all confidentiality levels and confidentiality groups added to the confidentiality matrix, or remove confidentiality group roles. Confidentiality group roles that were not deleted can be identified using the previously mentioned naming convention. No matter what you choose, the roles registered in the confidentiality group will not be deleted. The above example simply deletes the confidentiality matrix only.

7.4.6 Changing Confidentiality Level

Change the name and attributes of the confidentiality levels that were added to the confidentiality matrix 'matrix_foo' as follows. Specify the value after changing the name and attributes in JSON format. You can change the name or multiple attributes at once. Attributes not specified here are not changed.

```
select pgx_alter_confidential_level('matrix_foo', 'level1', '{"name": "levelX",
"comment":"This level required the highest confidential clearance."}')
```

If a confidentiality object was already registered with the confidentiality level, the attributes of the confidentiality object are also changed automatically. However, as mentioned in "7.2.1.1 Defining Confidentiality Levels", it may not be possible to automatically change the attributes of confidentiality objects. At that time, the function checks that the attributes of the registered confidentiality object are stricter than the attribute of the modified confidentiality level. If the check fails, this function will also fail. For example, if you try to change the encryption_algorithm from AES128 to AES256 and an AES128 encrypted table is registered as a confidentiality object, this function will fail. In such a case, please execute this function again after encrypting the table with AES256.

7.4.7 Deleting Confidentiality Level

Delete the confidentiality level as follows.

```
select pgx_drop_confidential_level('matrix_foo', 'level1', false)
```

With a third argument, you can choose to delete those that depend on the specified confidentiality level. Of course, confidentiality objects registered with the confidentiality level are not deleted.

7.4.8 Changing Confidentiality Group

Change the name and attributes of the confidentiality groups that were added to the confidentiality matrix 'matrix_foo' as follows. Specify the value after changing the name and attributes in JSON format. You can change the name or multiple attributes at once. Attributes not specified here are not changed.

```
select pgx_alter_confidential_group('matrix_foo', 'group1', '{"name": "groupX",
"comment":"Members of this group have the highest confidential clearance."}')
```

7.4.9 Deleting Confidentiality Group

Delete the confidentiality group as follows.

```
select pgx_drop_confidential_group('matrix_foo', 'group1', true, true)
```

By specifying a second or third argument, you can choose to delete dependencies on the specified confidentiality group or delete the confidentiality group role. Any choice will not delete the roles belonging to the confidentiality group.

Note

Even if you leave the confidentiality group role, this function will revoke privileges from the confidentiality group role. The privilege to revoke is the privilege defined in confidentiality privileges. Therefore, note that a role registered in a confidentiality group will be deprived of various privileges granted by inheriting the confidentiality group role.

7.4.10 Revoking Confidentiality Privileges

Revoke confidentiality privileges from confidentiality groups as follows. Specify the confidentiality object and confidentiality group and the privileges to revoke in JSON format. The following example revokes INSERT and DELETE privileges for table-type confidentiality objects belonging to 'level1' from 'group1'.

```
select pgx_revoke_confidential_privilege('matrix_foo','level1', 'group1',  '{"table":
["INSERT","DELETE"]}')
```

7.4.11 Removing Confidentiality Objects from Confidentiality Level

Remove confidentiality objects from the confidentiality level as follows. Specify the type and name of confidentiality objects to be excluded in JSON format. The following example removes table-type confidentiality objects schema1.table1 and schema1.table2 from the confidentiality level 'level1'.

```
select pgx_remove_object_from_confidential_level('matrix_foo', 'level1'
'[{
    "type":"table",
    "object":[
        {
          "schema":"schema1",
          "table":["table1","table2"]
        }
     ]
}]')
```

7.4.12 Removing Roles from Confidentiality Groups

Remove the role from the confidentiality group as follows. Specify the roles to remove in JSON format. The following example removes roles 'role1' and 'role2' from confidentiality group 'group1'.

```
select pgx_remove_role_from_confidential_group('matrix_foo', 'group1',
'["role1","role2"]')
```

7.5 Suggestions for Monitoring Methods

Confidentiality management role must ensure that the database is operating securely as intended, even after confidentiality managements are defined. If you are only using the confidentiality management feature, you do not have to worry about such things. However, the confidentiality management feature does not prohibit changing the definitions of tables and roles without using this feature.

So you have to detect when such an action has taken place.

However, even if they detect it, they may forget to deal with it. Therefore, it is necessary to periodically check the difference between the confidentiality level and confidentiality group and the actual definition of confidentiality objects and roles. Of course, even if there was a mismatch, it would not be a problem if the confidentiality object or role had stricter attributes and privilege.

The procedure presented here aims at matching.

7.5.1 How to Detect Privilege Changes without Using Confidentiality Management feature

Use the audit log to detect unauthorized modification of the attributes of confidentiality objects or roles, or modification of privileges without going through the confidentiality management feature.

The basic method for detection is to detect actions by roles other than the confidentiality management role. However, there are exceptions such as:

When performing an operation without using the confidentiality management feature for a legitimate reason

For example, when changing the authority of a function that the confidentiality management feature does not treat as a confidentiality object. In order to identify this, it is recommended to determine roles that perform changes that do not involve the confidentiality management feature. This is because when various roles do this, it becomes difficult to detect audit logs that deviate from operational rules.

When monitoring the activity of the confidentiality management role

For example, it would be a good idea to create rules that allow access only at specified times and from specified terminals, and to detect activities that violate those rules from audit logs. It is important to set rules so that violations cannot be covered up. For example application_name is not suitable as it can be easily spoofed.

7.5.2 How to Check Confidentiality Objects and Roles

This section describes how to confirm that the contents registered in the confidentiality matrix match the confidentiality objects and roles managed in the PostgreSQL system catalog. The method shown here is just an example.

Inspector

The inspector must be granted SELECT privilege on tables with the pg_confidential_management_support extension and SELECT privilege on the confidentiality object and role information added to the subject confidentiality matrix in the system tables. At a minimum, the confidentiality management role for the confidentiality matrix being reviewed and superuser certainly have such privileges.

Procedure

The confirmation procedure is divided into the following three phases. If you detect a discrepancy, use the audit log to investigate what caused the discrepancy. And fix it to match.

- Various objects may be changed (modified) in a chain reaction when returning to the correct state. Doing so may erase traces of unauthorized manipulation. Therefore, it is recommended that you investigate any design inconsistencies and ensure that you have the necessary audit logs for the investigation before fixing them.
- The *pgx_get_privileges_on_matrix* function presented here can output a very large table if the number of confidentiality objects or roles is large. If the size of this table exceeds the value of PostgreSQL's work_mem parameter, I/O will occur according to PostgreSQL's specifications, resulting in a slowdown. Therefore, it is recommended that work_mem be set as high as possible in the session in which this function is executed.

(1) Make sure the state of the confidentiality matrix is the same as the design

If you detect a mismatch, please restore the correct state by executing functions such as *pgx_alter_confidential_level* function provided by this extension.

- Attributes of the confidentiality matrix

  Check using the *pgx_confidential_matrix* table.

- Number of registered confidentiality levels and attributes of each confidentiality level

  Check the row that matches the *clevmatid* to be checked from the *pgx_confidential_level* table. To know the confidentiality matrix identifier, see *cmatid* in the *pgx_confidential_matrix* table.

- Number of registered confidentiality groups and attributes of each confidentiality group

  Check the row in the *pgx_confidential_role* table whose *ccolmatid* matches the identifier of the confidentiality matrix to be checked.

- Confidentiality group privileges set to confidentiality level

  Refer to the *pgx_confidential_privileges* table. However, in this table confidentiality levels and confidentiality groups are represented as identifiers. Join with *pgx_confidential_level table* and *pgx_confidential_role table* if you want to check by confidentiality level name or confidentiality group name.

(2) Verify correct confidentiality objects and roles registered

Confirm that the definition of the confidentiality management feature matches the definition of the confidentiality objects and roles. If you detect a mismatch, please restore the correct state by executing functions such as *pgx_alter_confidential_level* function provided by this extension.

- Is the number of confidentiality objects registered in the confidentiality level the same as designed?
- Do the attributes of each confidentiality object match the attributes of the confidentiality level to which it belongs?

  It is recommended to check these using *pgx_get_attribute_of_objects* function. Because this function gets the state of the confidentiality object from the PostgreSQL system catalog and outputs a table that can be compared with the definition. For the format of the table, refer to "B.7 Functions that Support Definition Referencing and Comparison with System Catalogs".

- Is the number of roles registered in the confidentiality group the same as designed?
- Do the attributes of each role match the attributes of the confidentiality group to which it belongs?

  It is recommended to check these using *pgx_get_attribute_of_roles* function.For the format of the table returned by a function that outputs authority information, such as this function, refer to "B.7 Functions that Support Definition Referencing and Comparison with System Catalogs".

(3) Check privilege of confidentiality objects

Verify that the roles granted access to confidentiality objects and what the privileges are are consistent with the definitions in the confidentiality matrix. A good help is to use the *pgx_get_privileges_on_matrix* function. For details on the output format, see "7.3.5 Granting Confidentiality Privileges to Confidentiality Groups".

If you want to focus on any confidentiality level or role, use the function below.

- pgx_get_privileges_on_level_and_group()
- pgx_get_privileges_on_object()
- pgx_get_privileges_on_role()

Point

It is inefficient to output a large table many times in order to analyze it with various SQL statements. You can perform efficient analysis by specifying a query that executes this function in the INSERT statement as shown below.

```
INSERT INTO temp_table_for_analysis SELECT pgx_get_privileges_on_matrix('matrix_foo')
```

7.6 Backup/Restore

There are 4 methods below.

- Use pg_basebackup
- Use pg_dumpall to back up the entire database cluster as a script file
- Use pg_dump and pg_restore to back up and restore only some databases
- Use pg_dump's pg_restore to back up and restore only some tables and schemas

Of these, if you adopt the method of using pg_basebackup, you do not need to be particularly careful. The following special precautions should be taken when adopting other methods.

Use pg_dumpall to back up the entire database cluster as a script file

Note the following.

- Do not use the -O option of the pg_dumpall command. This feature does not work properly if the confidentiality object changes ownership.
- Do not use the -x option of the pg_dumpall command. Of course, this would change tightly controlled privileges.
- Execute the backup script file as a superuser. A backup script file contains the CREATE EXTENSION statement for this extension. This is because the CREATE EXTENSION statement must be executed as a superuser.

Backup and restore only some databases using pg_dump and pg_restore

Note the following.

- Use the -r option of the pg_dumpall command to back up and restore role information as well. Of course, because roles are essential to confidentiality management.
- Do not use the -O option of the pg_dump command. This feature will not work properly if the confidentiality object changes ownership.
- Do not use the -x option of the pg_dump command. Of course, this would change tightly controlled privileges.
- Restore by a superuser for CREATE EXTENSION.

Use pg_dump's pg_restore to backup and restore only some tables and schema

Note the following.

- Use the -e option of the pg_dump command to back up and restore this extension as well. This is because the tables contained in this extension store the information required for confidentiality management.
- Use the -r option of the pg_dumpall command to back up and restore role information as well. Of course, because roles are essential to confidentiality management.
- Do not use the -O option of the pg_dump command. This feature will not work properly if the confidentiality object changes ownership.
- Do not use the -x option of the pg_dump command. Of course, this would change tightly controlled privileges.
- Restore by a superuser for CREATE EXTENSION.

7.7 Removing Setup

If you are no longer using this feature and want to continue using your database as before, we recommend simply DROP EXTENSION. This is because if you delete definitions such as confidentiality matrices and confidentiality levels created using this extension using the deletion functions provided by this extension, the roles automatically created by this extension will be deleted, the privileges of roles granted by this extension are revoked. Simply dropping this extension will only drop the tables it contains, it will not do any of these things.

7.8 Usage Example of Confidentiality Management

Here is an example of the concept of confidentiality levels and confidentiality groups.

This section assumes a simple business that handles customer purchase information.

First, create a confidentiality matrix for confidentiality management of this information.

```
SELECT pgx_create_confidential_matrix('matrix_purchase_managenet' , 'Confidentiality
management of customer purchase information');
```

The data we process may also contain personally identifiable information. Access to such data should be restricted to those who have access to it to minimize the risk of information disclosure.

Customer purchase information, including personally identifiable information, is managed in the following table.

```
CREATE TABLE purchase.customer_info(     -- Customer information
    customer_id     integer,               -- Customer ID
    name            text,
    address         text,
    phone_number    char(12),
    rank            integer                -- Customer's service rank
);

CREATE TABLE purchase.history(           -- History of a customer's purchase of goods
    customer_id         integer,            -- ID of the customer who purchased
    purchase_date       date,               -- Date the goods were purchased
    item_code           char(12),           -- Code of the goods purchased
    purchase_number     integer,            -- Quantity purchased
    purchase_amount     integer             -- Amount of purchased goods
);
```

Among customer information, name, address, and telephone number are personal information because they can identify an individual when combined. In order to properly handle such information, we have made it so that it can only be handled by employees belonging to a specific group who have received appropriate training.

Therefore, we have prepared two confidentiality levels: "level_personal_info", which means highly confidential personal information, and "level_customer_info", which means other information.

```
SELECT pgx_create_confidential_level('matrix_purchasee_management', 'level_personal_info',
                                     NULL, 'Personally identifiable information');
SELECT pgx_create_confidential_level('matrix_purchase_management', 'level_customer_info',
                                     NULL, 'Non-personally identifiable information');
```

In addition, we will prepare two confidentiality groups: "group_qualified " who have been educated about handling personal information and can handle personal information appropriately, and "group_non_qulified" who are not qualified.

```
SELECT pgx_create_confidential_group ('matrix_purchase_management', 'group_qualified',
                                   NULL, 'Qualified staff handling personal information');
SELECT pgx_create_confidential_group ('matrix_purchase_management', 'group_non_qulified',
                                     NULL, 'General employee');
```

Let's take a closer look at the data we're dealing with.

Since the customer information table contains personal information, it corresponds to personal information. However, the customer_id and rank contained in the customer information table are not personal information because they are not personally identifiable information. In addition, since this customer_id and rank are also information necessary for business analysis, it is inconvenient that only those who are qualified to handle personal information can access such information.

Therefore, the customer information table uses columns for confidentiality management. The entire customer information table is protected as personal information, and the range of access is expanded by making the columns that are not personal information general customer information.

Follow this policy to set confidentiality level privilege for confidentiality group.

```
SELECT pgx_grant_confidential_privilege('matrix_purchase_management',
                                     'level_personal_info',
                                     'group_qualified', '{"table":["ALL"]}');
SELECT pgx_grant_confidential_privilege('matrix_purchase_management',
                                     'level_customer_info',
                                     'group_qualified', '{"table":["ALL"]}');
SELECT pgx_grant_confidential_privilege('matrix_purchase_management',
                                     'level_customer_info',
```

```
                                        'group_not_qualified',
                                        '{"table":["ALL"], "column":["SELECT"]}');
```

Only "qualified personnel" can handle "personal information". "Customer information" can be handled by both "qualified personnel" and "general employees". Some columns of tables that handle "personal information" are allowed to be referred to as "customer information".

This completes the authorization settings in the confidentiality matrix.

Next, we will register the database objects that handle purchase information in the confidentiality matrix.

```
SELECT pgx_add_object_to_confidential_level('matrix_purchase_management',
'level_personal_info',
                                    '[{
                                         "type":"table",
                                         "object":[{
                                                      "schema": "purchase",
                                                      "table": ["customer_info"]
                                          }]
                                     }]');
SELECT pgx_add_object_to_confidential_level('matrix_purchase_management',
'level_customer_info',
                                    '[{
                                         "type":"column",
                                         "object":[{
                                                       "schema": "purchase",
                                                       "table": "customer_info",
                                                       "column": ["customer_id", "rank"]
                                          }]
                                     }]');
SELECT pgx_add_object_to_confidential_level('matrix_purchase_management',
'level_customer_info',
                                    '[{
                                         "type":"table",
                                         "object":[{
                                                       "schema": "purchase",
                                                       "table": ["history"]
                                     }]
                                     }]');
```

The entire customer information table is "personal information", the customer_id column and rank column of the customer information table are "customer information", and the entire purchase history table is also "customer information".

Finally, enroll the employee in the confidentiality group. "Alex" and "Bola" are "qualified persons" who have received training in personal information management. Also, "Charlie" and "Dana" are "general employees" because they have not yet received training on personal information management.

```
SELECT pgx_add_role_to_confidential_group('matrix_purchase_management',
                                          'group_qualified',
                                          '["Alex", "Bola"]');
SELECT pgx_add_role_to_confidential_group('matrix_purchase_management',
                                          'group_non_qualified',
                                          '["Charlie", "Dana"]');
```

Appendix A Tables Used by Confidentiality Management Feature

This section describes the tables used by the confidentiality management feature.

A.1 pgx_confidential_matrix

A list of confidentiality matrices.

You can refer to the attributes of the registered confidentiality matrix, the update time, or the time when the confidentiality level or confidentiality group was registered or deleted.

Column name	Type	Constraint	Description
cmatid	bigint	primary key generated always as identity	Identifier of the confidentiality matrix.
cmatname	varchar(63)	unique not null	Name of the confidentiality matrix.
cmatowner	name	not null	Owner of the confidentiality matrix. The role that created the confidentiality matrix becomes the owner.
cmatcomment	text		Comment.
cmatupdatetime	timestamp with time zone	not null	Update time of the confidentiality matrix itself.
cmatoperationtime	timestamp with time zone		The time when the confidentiality level and confidentiality group were added/deleted.

A.2 pgx_confidential_level

A list of confidentiality levels.

You can refer to the registered confidentiality level attributes, update time, or the time when a confidentiality object was registered to the confidentiality level or removed from the confidentiality level.

Column name	Type	Constraint	Description
clevid	bigint	primary key generated always as identity	Identifier of the confidentiality level.
clevname	varchar(63)	not null	Name of the confidentiality level.
clevmatid	bigint	not null references pgx_confidential_matrix(cmatid)	Identifier of the confidentiality matrix to which the confidentiality level belongs.
clevcomment	text		Comment.
clevupdatetime	timestamp with time zone	not null	Update time of the confidentiality level itself.

Column name	Type	Constraint	Description
clevoperationtime	timestamp with time zone		The time when the confidentiality object was added/deleted.
clevencalgorithm	text	not null	Encryption method. "none" for no encryption. AES128 and AES256 can be set.

A.3 pgx_confidential_group

A list of confidentiality groups.

You can refer to the registered confidentiality group attributes, update time, or the time when a role was registered to the confidentiality group or removed from the confidentiality group.

Column name	Type	Constraint	Description
cgroid	bigint	primary key generated always as identity	Identifier of the confidentiality group.
cgroname	varchar(63)	not null	Name of the confidentiality group.
cgromatid	bigint	not null references pgx_confidential_matrix(cmatid)	Identifier of the confidentiality matrix to which the confidentiality group belongs.
cgrocomment	text		Comment.
cgroupdatetime	timestamp with time zone	not null	Update time of the confidentiality group itself.
cgrooperationtime	timestamp with time zone		The time when the role was added/deleted.
cgrorolename	name	not null	Name of the confidentiality group role.
cgrosuperuser	bool	not null	true if the confidentiality group role has SUPERUSER privileges.
cgrocreatedb	bool	not null	true if the confidentiality group role has CREATEDB privileges.
cgrocreaterole	bool	not null	true if the confidentiality group role has CREATEROLE privileges.
cgroreplication	bool	not null	true if the confidentiality group role has REPLICATION privileges.
cgrobypassrls	bool	not null	true if the confidentiality group role has BYPASSRLS privileges.

A.4 pgx_confidential_privilege

A list of confidentiality privileges.

You can refer to confidentiality privilege set for each confidentiality object, update time, and so on.

Column name	Type	Constraint	Description
cpriid	bigint	primary key generated always as identity	Identifier of the privilege.

Column name	Type	Constraint	Description
cprimatid	bigint	not null references pgx_confidential_matrix(cmatid)	Identifier of the confidentiality matrix to which the privilege belongs.
cprilevid	bigint	not null references pgx_confidential_level(clevid)	Identifier of the confidentiality level for which privilege is set.
cprigroid	bigint	not null references pgx_confidential_group(cgroid)	Identifier of the confidentiality group for which privilege is set.
cpritype	text	not null	Type of the confidentiality object which privilege is set.
cpriupdatetime	timestamp with time zone	not null	Update time when privilege was set/ changed.
cpriacl	text[]	not null	Access privileges that have been set. (*1)

*1: The character string indicating authority appears in the following order in the text type array of cpriacl.

ALL, SELECT, INSERT, UPDATE, DELETE, TRUNCATE, REFERENCES, TRIGGER, CREATE, CONNECT, TEMPORARY, EXECUTE, USAGE

If the appropriate privilege is not set, the string simply does not appear. For example, {'INSERT','TRUNCATE'} if you only have INSERT and TRUNCATE privileges.

A.5 pgx_confidential_object

A list of confidentiality objects.

You can refer to object attributes or update time, and so on.

Column name	Type	Constraint	Description
cobjid	bigint	primary key generated always as identity	Identifier of the confidentiality object.
cobjmatid	bigint	not null references pgx_confidential_matrix(cmatid)	Identifier of the confidentiality matrix to which the confidentiality object belongs.
cobjlevid	bigint	not null references pgx_confidential_level(clevid)	Identifier of the confidentiality level to which the confidentiality object belongs.
cobjtype	text	not null	The type of confidentiality object.
cobjschema	name	not null	Schema name of the confidentiality object.
cobjtable	name	not null	Table name of the confidentiality object.
cobjname	text	not null	Name of the confidentiality object.

Column name	Type	Constraint	Description
cobjupdate	timestamp with time zone	not null	Registration time of the confidentiality object.
cobjpolicy	jsonb		Conditions that determine the range of rowsets when the type is rowset. It is expressed by the setting contents in POLICY.

A.6 pgx_confidential_role

A list of roles registered in the confidentiality group.

You can refer to role attributes or update time, and so on.

Column name	Type	Constraint	Description
crolid	bigint	primary key generated always as identity	Identifier of the role.
crolmatid	bigint	not null references pgx_confidential_matrix(cmatid)	Identifier of the confidentiality matrix to which the role belongs.
crolgroid	bigint	not null references pgx_confidential_group(cgroid)	Identifier of the confidentiality group to which the role belongs.
crolname	name	not null	Name of the role.
crolupdate	timestamp with time zone	not null	Registration time of the role.

A.7 pgx_confidential_policy

This is a list of policies created to set privileges for confidentiality objects of rowset type. You can refer to the name of the policy you created and the privileges it has set.

Rows in this table are inserted when you add a rowset type confidentiality object.

Column name	Type	Constraint	Description
cpolid	bigint	primary key generated always as identity	Identifier of the policy.
cpolmatid	bigint	not null references pgx_confidential_matrix(cmatid)	Identifier of the confidentiality matrix to which the policy belongs.
cpollevid	bigint	not null references pgx_confidential_level(clevid)	Identifier of the confidentiality level to which the policy belongs.
cpolgroid	bigint	not null references	Identifier of the confidentiality group to which the policy belongs.

Column name	Type	Constraint	Description
		pgx_confidential_group(cgroid)	
cpolobjid	bigint	not null references pgx_confidential_object(cobjid)	Identifier of the rowset object using this policy.
cpolprivilege	text	not null	Privilege this policy has (SELECT, INSERT, UPDATE, DELETE, ALL).
cpolname	name	not null	Name of the policy.
cpolexpression	jsonb	not null	Expression of the policy.

Appendix B System Management Functions Used by Confidentiality Management Feature

This section describes the system management functions used by the confidentiality management feature. All functions abort the transaction on failure.

Note

- Be careful when performing operations that involve deleting confidentiality groups.

  If you remove a confidentiality group along with a confidentiality group role, you simply no longer have the role that can access the confidentiality object. However, when you leave the confidentiality group role, the function revokes privileges from the confidentiality group role. The privilege to revoke is the privilege defined in confidentiality privileges.

- The *pgx_get_privileges_on_matrix* function may output a very large table if the number of confidentiality objects or roles is large. If the size of this table exceeds the value of PostgreSQL's *work_mem* parameter, I/O will occur according to PostgreSQL's specifications and will be slow. To prevent this, it is recommended that *work_mem* be set as high as possible in the session in which this function is executed.

B.1 Confidentiality Matrix Manipulation Functions

Function name	Return value	Description
pgx_create_confidential_matrix(confidential_matrix_name varchar, comment text)	void	Create the confidentiality matrix with the specified name. The created confidentiality matrix is registered in the *pgx_confidential_matrix* table along with the comment. Only roles with the required entitlements for the confidentiality management role can execute this function. The role that executes this function is the confidentiality management role for the confidentiality matrix. Functions such as executing by specifying a confidentiality matrix name require that the executed role is a confidentiality management role for the specified confidentiality matrix. For details, please refer to "7.2.2 Determining Confidentiality Management Roles". The length of *confidential_matrix_name* must be less than 64 characters. Note that the units are not bytes. There are no restrictions on the characters that can be used in the *confidential_matrix_name*. When you specify the name of the confidentiality matrix to other functions, you must specify the same string that you specified to this function. Note that unlike most CREATE statements, the name of the confidentiality matrix is case sensitive.
pgx_copy_confidential_matrix(confidential_matrix_name varchar, source_confidential_matrix_name varchar)	void	Copy the source confidentiality matrix specified by *source_confidential_matrix_name* to the confidentiality matrix named *confidential_matrix_name*. Confidentiality matrix, confidentiality levels, confidentiality groups, and confidentiality privileges details are replicated. However, the information of confidentiality objects registered in the confidentiality level and roles registered in the confidentiality group are not duplicated.

Function name	Return value	Description
		Any confidentiality management role in any confidentiality matrix can execute this function. The owner of the cloned confidentiality matrix is the role that executed this function. Restrictions on *confidential_matrix_name* as strings are the same as for the *pgx_create_confidential_matrix* function. Comments are also duplicated.

pgx_alter_confidential_matrix(confidential_matrix_name varchar, alter_object json)	void	Change the attributes of the confidentiality matrix named by *confidential_matrix_name*.
		
		Only confidentiality management role for the specified confidentiality matrix can execute this function.
		
		For *alter_object*, specify the attribute you want to change and the value after change in key-value format as follows. Attributes not specified remain unchanged.
		
		`'{`
		`  "name":"matrix_foo",`
		`  "comment":"This matrix is defined for foo.")`
		`}'`
		
		name: Specify the name of the modified confidentiality matrix. Cannot be *null*. The function will fail if you specify the name of a confidentiality matrix that already exists.
		
		comment: Specify a comment after the change. Can be *null*.

pgx_drop_confidential_matrix(confidential_matrix_name varchar, cascade bool, drop_role bool)	void	Drop the confidentiality matrix with the specified name. Only confidentiality management role for the specified confidentiality matrix can execute this function. Specify *true* for cascade to recursively check and remove objects that depend on this confidentiality matrix. For example, delete the confidentiality groups and confidentiality levels registered in this confidentiality matrix. Then remove the confidentiality privileges associated with that confidentiality level. To drop a confidentiality level, execute internally *pgx_drop_confidential_level* function with a *cascade* value. When dropping a confidentiality group, execute internally *pgx_drop_confidential_group* function with *cascade* and *drop_role* values. See also the descriptions of these functions. In particular, how the privilege of confidentiality objects are changed is important. Specify *false* for cascade simply removes the confidentiality matrix. The function will fail if there are objects that depend on this confidentiality matrix. If *true* is specified for *drop_role*, the confidentiality group role registered in this confidentiality matrix will be deleted. Naturally, it only makes sense when *cascade* is *true*.

B.2 Confidentiality Level Manipulation Functions

Function name	Return value	Description

pgx_create_confidential_level(confidential_matrix_name varchar, confidential_level_name varchar, options json, comment text)	void	Creates a confidentiality level, registers it with the specified confidentiality matrix, and adds it to the *pgx_confidential_level* table with the specified comment and attributes specified in *options*.
		
		Only confidentiality management role for the specified confidentiality matrix can execute this function.
		
		The length of *confidential_level_name* must be less than 64 characters.
		
		Note that the units are not bytes.
		
		There are no restrictions on the characters that can be used in the *confidential_level_name*.
		
		When specifying a confidentiality level name for any other function, you must specify the same string as specified for this function
		
		Note that unlike most CREATE statements, confidentiality level names are case-sensitive.
		
		Specify a comment for *comment*.
		
		For *options*, specify the attribute of the confidentiality level as follows. If you specify NULL, the default value for each attribute will be set.
		
		`'{`
		`  "encryption_algorithm":"AES256"`
		`}'`
		
		encryption_algorithm: Specify the encryption algorithm. The algorithms and default values that can be specified are the same as the *tablespace_encryption_algorithm* parameter of Transparent Data Encryption of Fujitsu Enterprise Postgres. Must not be *null*.

pgx_alter_confidential_level(confidential_matrix_name varchar, confidential_level_name varchar, alter_object json)	void	Change the confidentiality level attribute.
		
		Only confidentiality management role for the specified confidentiality matrix can execute this function.
		
		For *alter_object*, specify the attribute you want to change and the value after change in key-value format as follows.
		
		`'{`
		`  "name":"level_new",`
		`  "comment":"This level is the highest confidentiality level.",`
		`  "encryption_algorithm":"AES256"`
		`}'`
		
		name: Specify the name of the modified confidentiality level. Cannot be null.
		
		comment: Specify a comment after the change. Can be null.
		
		Other attributes are the same as options of *pgx_create_confidential_level* function. Attributes not specified are not changed.
		
		Be careful when increasing the degree of confidentiality. For example, if you increase the encryption strength and

Function name	Return value	Description
		there are confidentiality objects with a lower strength than the new strength, this function will fail.
pgx_drop_confidential_level(confidential_matrix_name varchar, confidential_level_name varchar, cascade bool)	void	Remove a confidentiality level from the confidentiality matrix and delete a confidentiality level. Only confidentiality management role for the specified confidentiality matrix can execute this function. If *true* is specified for *cascade*, the confidentiality level can be deleted even if confidentiality objects are registered in this confidentiality level. If *false* is specified for *cascade*, it is not possible to delete a confidentiality level that has confidentiality objects registered.

B.3 Confidentiality Group Manipulation Functions

Function name	Return value	Description

pgx_create_confidential_group(confidential_matrix_name varchar, confidential_group_name varchar, options json, comment text)	void	Create the confidentiality group, registers it with the specified confidentiality matrix, and adds it to the *pgx_confidential_group* table with the specified comment and attributes specified in *options*.
		Only confidentiality management role for the specified confidentiality matrix can execute this function. However, setting some attributes requires superuser privileges, as described below. Therefore, by necessity, the confidentiality management role must be superuser if those attributes are to be set.
		This function internally uses the CREATE ROLE statement to create a confidentiality group role.
		The length of *confidential_group_name* must be less than 64 characters. Note that the units are not bytes.
		There are no restrictions on the characters that can be used in the *confidential_group_name*.
		When you specify the name of the confidentiality group to other functions, you must specify the same string that you specified to this function.
		Note that unlike most CREATE statements, confidentiality group names are case-sensitive.
		For *options*, specify the attributes of the confidentiality group as follows. If you specify NULL, the default value for each attribute will be set.
		`'{`
		`  "SUPERUSER":false,`
		`  "CREATEDB":true,`
		`  "CREATEROLE":false,`
		`  "REPLICATION":false,`
		`  "BYPASSRLS":false`
		`}'`

Function name	Return value	Description
		The only attributes that can be specified are the SUPERUSER, CREATEDB, CREATEROLE, REPLICATION, and BYPASSRLS attributes that relate to access privileges to data. These attributes are some of the role attributes that can be specified in the CREATE ROLE statement. The attribute semantics and default values are the same as in the CREATE ROLE statement specification. As noted in the CREATE ROLE statement description, this function fails if a non-superuser specifies *true* for the SUPERUSER, REPLICATION, and BYPASSRLS attributes.

pgx_alter_confidential_group(confidential_matrix_name varchar, confidential_group_name varchar, alter_object json);	void	Change the attributes of a confidentiality group.
		
		Only confidentiality management role for the specified confidentiality matrix can execute this function. As described in *pgx_create_confidential_group* function, you must be superuser to set some attributes.
		
		This function internally uses the ALTER ROLE statement to change the attributes of the confidentiality group role.
		
		For *alter_object*, specify the attribute you want to change and the value after change in key-value format as follows.
		
		`'{`
		`  "name":"group_new",`
		`  "comment":"Members of this group have the highest confidential clearance.",`
		`  "CREATEDB":false`
		`}'`
		
		name: Specify the name of the confidentiality group after modification. Cannot be null.
		
		comment: Specify a comment after the change. Can be null.
		
		Other attributes are the same as options of *pgx_create_confidential_group* function. Attributes not specified are not changed.
		
		For example, if you change the attribute to a weaker one, such as changing CREATEDB to *false*, the attributes of roles registered in the confidentiality group will be similarly weakened.

pgx_drop_confidential_group(confidential_matrix_name varchar, confidential_group_name varchar, cascade bool, drop_role bool)	void	Drop the confidentiality group from the confidentiality matrix and delete a confidentiality group. Only confidentiality management role for the specified confidentiality matrix can execute this function. If *true* is specified for cascade, the confidentiality group can be deleted even if roles are registered in this confidentiality group. If *false* is specified for *cascade*, confidentiality groups that have roles registered cannot be deleted. Specify true for *drop_role* to drop the confidentiality group role. Roles registered in confidentiality groups remain. If

Function name	Return value	Description
		false is specified for *drop_role*, the confidentiality group role will not be deleted. When you leave the confidentiality group role, the function revokes privileges from the confidentiality group role. The privilege to revoke is the privilege defined in confidentiality privileges.

B.4 Confidentiality Privilege Manipulation Functions

Function name	Return value	Description

pgx_grant_confidential_privilege(confidential_matrix_name varchar, confidential_level_name varchar, confidential_group_name varchar, privilege json)	void	Grant confidentiality privileges.
		
		Only confidentiality management role for the specified confidentiality matrix can execute this function.
		
		Grants access to the confidentiality level specified by *confidential_level_name* to the confidentiality group specified by *confidential_group_name*.
		
		When you run this function repeatedly, it simply adds more privileges to grant. Granting the same privilege more than once does not result in an error.
		
		The privilege to be granted is specified in *privilege*. *privilege* specifies the type of the confidentiality object as the key and an array of privileges as the value, like this:
		
		`'{`
		`  "table":["SELECT", "INSERT", "UPDATE", "DELETE"],`
		`  "schema":["CREATE", "USAGE"],`
		`  "rowset":["ALL"]`
		`}'`
		
		The privileges that can be specified depend on the type of confidentiality object.
		
		Privilege for rowset type confidentiality object is privilege that can be specified in the FOR clause of the CREATE POLICY statement.
		
		Except for the rowset type, it is a privilege that can be granted with the GRANT statement according to the confidentiality object type.
		
		If ALL is specified, it is assumed that all privileges that can be specified for that type are listed.
		
		That is, ALL does not appear in the *cpriacl* column of the *pgx_confidential_privilege* table.
		
		The same specification as the WITH GRANT OPTION clause of the GRANT statement cannot be specified. This is because only confidentiality management roles should use this feature to change privilege to confidentiality objects.
		
		Be carefull when PUBLIC is granted to target confidentiality object. This is because granting privileges to PUBLIC is the same as granting privileges to all roles

Function name	Return value	Description
		registered in the confidentiality matrix. This function will fail if a privilege granted indirectly to each role using PUBLIC is defined in the confidentiality privileges that should not be granted to that role. Similarly, this function also checks privileges granted indirectly through group roles that are not registered in the confidentiality matrix. In doing so, it recursively checks the chain of inheritance.
pgx_revoke_confidential_privilege(confidential_matrix_name varchar, confidential_level_name varchar, confidential_group_name varchar, privilege json)	void	Revoke confidentiality privileges. Only confidentiality management role for the specified confidentiality matrix can execute this function. Revokes privilege to the confidentiality level specified by *confidential_level_name* from the confidentiality group specified by *confidential_group_name*. Revoking ungranted privileges does not fail. Privilege to be revoked is specified in *privilege*. The specification method is the same as *pgx_grant_confidential_privilege* function. Be carefull when PUBLIC is granted to target confidentiality object. This is because granting privileges to PUBLIC is the same as granting privileges to all roles registered in the confidentiality matrix. This function will fail if a privilege granted indirectly to each role using PUBLIC is defined in the confidentiality privileges that should not be granted to that role. Similarly, this function also checks privileges granted indirectly through group roles that are not registered in the confidentiality matrix. This time, it checks the chain of inheritance up to its ancestors.

B.5 Confidentiality Object Manipulation Functions

Function name	Return value	Description
pgx_add_object_to_confidential_level(confidential_matrix_name varchar, confidential_level_name varchar, object_name json)	void	Only confidentiality management role for the specified confidentiality matrix can execute this function. Adds the confidentiality object specified by *object_name* to the confidentiality level specified by *confidential_level_name*. This function internally uses the GRANT statement to grant privileges to the confidentiality group role according to the confidentiality privileges associated with this confidentiality level. However, if the confidentiality object is of type rowset, it internally uses the CREATE POLICY statement to grant privileges to the confidentiality group role according to the confidentiality privileges associated with this confidentiality level. Also, to enable POLICY, execute the ALTER TABLE statement on the target table with the ENABLE ROW LEVEL SECURITY clause.

Function name	Return value	Description

		Currently it is not possible to register a foreign table as a table type confidentiality object.
		
		Be carefull when PUBLIC is granted to target confidentiality object. This is because granting privileges to PUBLIC is the same as granting privileges to all roles registered in the confidentiality matrix. This function will fail if a privilege granted indirectly to each role using PUBLIC is defined in the confidentiality privileges that should not be granted to that role. Similarly, this function also checks privileges granted indirectly through group roles that are not registered in the confidentiality matrix. In doing so, it recursively checks the chain of inheritance.
		
		For rowset type confidentiality objects, this function will fail if the target table has a POLICY defined that was not created using this feature, regardless of what privileges are granted.
		
		These checks only apply to the confidentiality group role or to roles that are registered with the confidentiality group. If a POLICY exists that targets a role that is not, the function will not fail.
		
		Specify object_name as follows: Only the rowset is slightly different. The example below attempts to register multiple types of objects in one go.
		
		`'[{`
		`  "type":"schema",`
		`  "object":[`
		`  {"schema":"schema1"},`
		`  {"schema":"schema2"}`
		`  ]`
		`},`
		
		`{`
		`  "type":"table",`
		`  "object":[`
		`  {`
		`  "schema":"schema1",`
		`  "table":["table1","table2"]`
		`  },`
		`  {`
		`  "schema":"schema2",`
		`  "table":["table8","table9"]`
		`  }`
		`  ]`
		`},`
		
		`{`
		`  "type":"column",`
		`  "object":[`
		`  {`
		`  "schema":"schema1",`
		`  "table":"table1",`
		`  "column":["column1", "column2"]`
		`  },`
		`  {`
		`  "schema":"schema1",`
		`  "table":"table2",`

Function name	Return value	Description

		<pre>"column":["column8", "column9"]
		}
		]
		}]'</pre>
		For the rowset type, you define rowset and give it a name, as in the example below. This name is used by the *pgx_remove_object_from_confidential_level* function to identify the rowset type confidentiality object when removing it.
		
		This example shows:
		
		- In schema1.table1, represents a set of rows (rowset) for which the conditional expression (user = current_user OR manger = current_user) is true.
		- If manager=current_user is true, the col1 value of the record to be INSERTed or the record after UPDATE must be greater than zero.
		
		<pre>'[{
		"type":"rowset",
		"object":[
		{
		"schema":"schema1",
		"table":"table1",
		"rowset_name":"rowset1",
		"rowset_expression": [
		{
		"as":"permissive",
		"using":"user = current_user"
		},
		{
		"as":"permissive",
		"using":"manager =
		current_user",
		"with check":"col1 > 0"
		}
		]
		}
		]
		}]'</pre>
		Each key (as, *using*, *with check*) has the same meaning as the clause of the same name in the CREATE POLICY statement.
		
		As you can see from this, one element of the array specified in *rowset_expression* corresponds to one POLICY object created by the CREATE POLICY statement.
		
		In fact, this function internally executes as many CREATE POLICY statements as there are elements in the array. The name of POLICY at this time is 'pgx_cms_policy_${cpolid}'.
		
		${cpolid} is automatically numbered by this extension.
		
		<Note>

Function name	Return value	Description
		Do not create policies with names that begin with *pgx_cms_policy_*. Because this function may fail.
pgx_remove_object_from_confidential_level(confidential_matrix_name varchar, confidential_level_name varchar, object_name json)	void	Removes confidentiality objects from the confidentiality level. Only confidentiality management role for the specified confidentiality matrix can execute this function. Removes the confidentiality object specified by *object_name* from the confidentiality level specified by *confidential_level_name*. The format of *object_name* is the same as *object_name* in the *pgx_add_object_to_confidential_level* function. But the *rowset_expression* key is optional. If specified, it is simply ignored. Therefore, *object_name* in *pgx_add_object_to_confidential_level* can be specified to this function without modifying it. At this time, any privileges granted to the confidentiality group based on confidentiality privileges are revoked. If the confidentiality object is of rowset type, delete the internally created policy. At this time, if there are zero policies associated with the table, the ALTER TABLE statement with the DISABLE ROW SECURITY clause is internally executed to disable row level security.

B.6 Role Manipulation Functions

Function name	Return value	Description
pgx_add_role_to_confidential_group(confidential_matrix_name varchar, confidential_group_name varchar, role_name json)	void	Add a role to a confidentiality group. Only confidentiality management role for the specified confidentiality matrix can execute this function. Adds the role specified by *role_name* to the confidentiality group specified by *confidential_group_name*. If the role to be added has been granted broader privileges than the confidentiality privileges, revoke the privileges according to the confidentiality privileges. Be carefull when PUBLIC is granted to target confidentiality object. This is because granting privileges to PUBLIC is the same as granting privileges to all roles registered in the confidentiality matrix. This function will fail if a privilege granted indirectly to each role using PUBLIC is defined in the confidentiality privileges that should not be granted to that role. Similarly, this function also checks privileges granted indirectly through group roles that are not registered in the confidentiality matrix. In doing so, it recursively checks the chain of inheritance.

Function name	Return value	Description
		Also, if the added role has stronger attributes than the confidentiality group, change the attributes to match the confidentiality group. This function will fail if a strong attribute is indirectly assigned using a group role that is not registered in the confidentiality matrix. *role_name* is specified as follows. '{["role1","role"]}'
pgx_remove_role_from_confidential_group(confidential_matrix_name varchar, confidential_group_name varchar, role_name json)	void	Remove a role from a confidentiality group. Only confidentiality management role for the specified confidentiality matrix can execute this function. Removes the role specified by *role_name* from the confidentiality group specified by *confidential_group_name*. This function internally executes a REVOKE statement to remove the role from the confidentiality group role. It does not change the attributes of the role being removed, nor the privileges granted to that role. Simply banish it from the group so that it cannot inherit privileges. The method of specifying role_name is the same as *pgx_add_role_to_confidential_group* function.

B.7 Functions that Support Definition Referencing and Comparison with System Catalogs

Functions that Support Definition Referencing and Comparison with System Catalogs

Function name	Return value	Description
pgx_get_attribute_of_objects(confidential_matrix_name varchar)	setof record	Returns a table of attributes defined in the confidentiality matrix and attributes actually set in the database for all confidentiality objects registered in the specified confidentiality matrix. Refer to "Tables returned by pgx_get_attribute_of_objects" for the table format. Only confidentiality management role for the specified confidentiality matrix can execute this function.
pgx_get_attribute_of_roles(confidential_matrix_name varchar)	setof record	Returns a table of attributes defined in the confidentiality matrix and attributes actually set in the system catalog for all roles registered in the specified confidentiality matrix. Refer to "Table returned by pgx_get_attribute_of_roles" for the format of the table. Only confidentiality management role for the specified confidentiality matrix can execute this function.
pgx_get_privileges_on_level_and_group(confidential_matrix_name varchar, confidential_level_name	setof record	Returns a table that allows you to compare the following for combinations of confidentiality objects registered in the specified confidentiality level and roles registered in the

Function name	Return value	Description
varchar, confidential_group_name varchar)		specified confidentiality group. Refer to "Table returned by pgx_get_privileges_on_level_and_group" for the format of the table. - Privileges specified by the confidentiality privilege settings that should be granted to the specified role - Privileges granted in the actual system catalog Only confidentiality management role for the specified confidentiality matrix can execute this function.
pgx_get_privileges_on_object(confidential_matrix_name varchar, object_name json)	setof record	Returns a table that allows you to compare the following for the specified confidentiality objects. Refer to "Table returned by pgx_get_privileges_on_object" for the format of the table. - Privileges dictated by confidentiality privilege settings that should be granted to all roles - Privileges granted in the actual system catalog Only confidentiality management role for the specified confidentiality matrix can execute this function.
pgx_get_privileges_on_role(confidential_matrix_name varchar, role_name json)	setof record	Returns a table that allows you to compare the following for the all confidentiality objects. Refer to "Table returned by pgx_get_privileges_on_role" for the format of the table. - Privileges specified by the confidentiality privilege settings that should be granted to the specified role - Privileges granted in the actual system catalog Only confidentiality management role for the specified confidentiality matrix can execute this function.
pgx_get_privileges_on_matrix(confidential_matrix_name varchar)	setof record	Returns a table that allows you to compare the following for the all confidentiality objects in the specified confidentiality matrix. Refer to "Table returned by pgx_get_privileges_on_matrix" for the format of the table. - Privileges defined by confidentiality privilege settings that should be granted to all roles registered in the confidentiality matrix. - Privileges granted in the actual system catalog Only confidentiality management role for the specified confidentiality matrix can execute this function.

Tables returned by pgx_get_attribute_of_objects

Column name	Type	Description
matrix_name	varchar(63)	Confidentiality matrix name
confidential_level_name	varchar(63)	Confidentiality level name
object_type	text	Confidentiality object type
object_schema	name	Confidentiality object schema name

Column name	Type	Description
object_table	name	Confidentiality object table name
object_name	text	Confidentiality object name
rowset_expression	json	Conditional expression when the confidentiality object is a row
encrypt_on_matrix	text	Encryption method and strength specified in the confidentiality matrix
encrypt_on_object	text	Actual encryption method and strength of confidentiality objects

Tables returned by pgx_get_attribute_of_roles

Column name	Type	Description
matrix_name	varchar(63)	Confidentiality matrix name
confidential_group_name	varchar(63)	Confidentiality group name
role_name	name	Role name
confidential_group_role	bool	Indicates whether it is a confidentiality group role or not. *true* if it is a confidentiality group role
superuser_on_matrix	bool	SUPERUSER attribute specified in the confidentiality matrix
superuser_on_role	bool	Actual SUPERUSER attribute of the role
createdb_on_matrix	bool	CREATEDB attribute specified in the confidentiality matrix
createdb_on_role	bool	Actual CREATEDB attribute of the role
createrole_on_matrix	bool	CREATEROLE attribute specified in the confidentia matrix
createrole_on_role	bool	Actual CREATEROLE attribute of the role
replication_on_matrix	bool	REPLICATION attribute specified in the confidentia matrix
replication_on_role	bool	Actual REPLICATION attribute of the role
bypassrls_on_matrix	bool	BYPASSRLS attribute specified in the confidentia matrix
bypassrls_on_role	bool	Actual BYPASSRLS attribute of the role

Tables returned by pgx_get_privileges_on_level_and_group, pgx_get_privileges_on_object, pgx_get_privileges_on_role and pgx_get_privileges_on_matrix

Column name	Type	Description
matrix_name	varchar(63)	Confidentiality matrix name
confidential_level_name	varchar(63)	Confidentiality level name

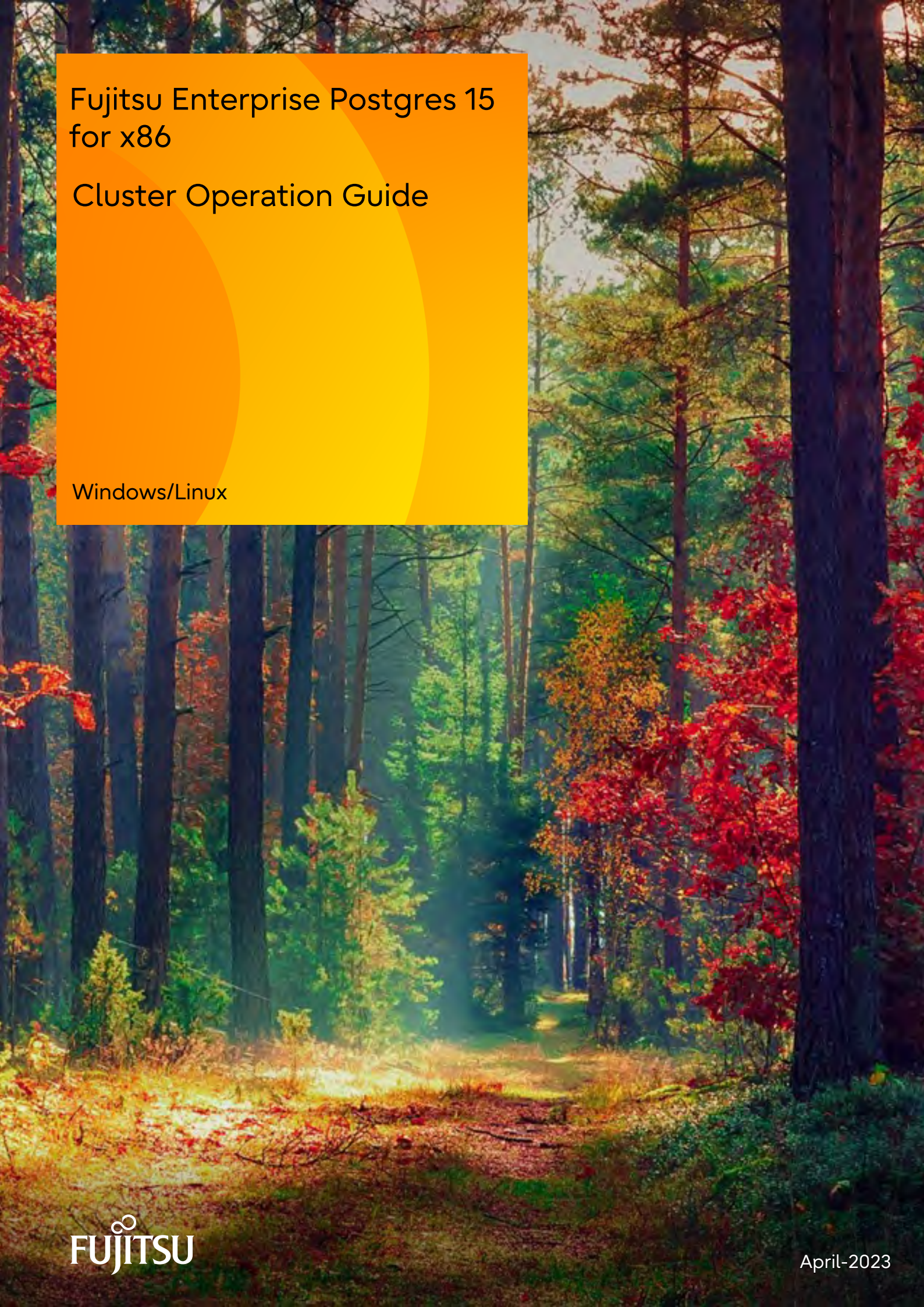
Column name	Type	Description
confidential_group_name	varchar(63)	Confidentiality group name
object_type	text	Confidentiality object type
object_scheme	name	Confidentiality object schema name
object_table	name	Confidentiality object table name
object_name	text	Confidentiality object name
role_name	name	Role name
privilege_list_on_matrix	text[]	Privileges specified by the confidentiality matrix settings. Output is separated by commas
privilege_list_on_object	text[]	Privileges specified by the confidentiality matrix settings. Output is separated by commas
policy_name	name	NULL if the confidentiality object is not of type rowset In the case of rowset type, rowset name (*1)
policy_setting_on_matrix	jsonb	NULL if the confidentiality object is not of type rowset In the case of rowset type, the rowset policy information set in the confidentiality matrix (*1)
policy_setting_on_policy	jsonb	NULL if the confidentiality object is not of type rowset In case of rowset type, row policy information set in the actual policy (*1)

*1: When adding a rowset type confidentiality object, multiple privileges can be set at once, which is not represented by a single row in this table. For example, if you set SELECT and DELETE privileges, you will see a row for SELECT privileges and a row for DELETE privilege. This is because rowset type access control uses PostgreSQL's row-level security POLICY. In this specification, POLICY for SELECT privilege is different from POLICY for DELETE privilege.

Fujitsu Enterprise Postgres 15 for x86

Cluster Operation Guide

Windows/Linux



FUJITSU

April-2023

Database Multiplexing (Linux) >
Database Multiplexing (Windows) >
PRIMECLUSTER >

Fujitsu Enterprise Postgres 15 for x86

Cluster Operation Guide

Database Multiplexing

Linux

FUJITSU

J2UL-2842-01ENZ0(00)
April-2023

Preface

Purpose of this document

This document describes the tasks required for using the database multiplexing feature of Fujitsu Enterprise Postgres.

Intended readers

This document is intended for those who set up and use the database multiplexing feature.

Readers of this document are also assumed to have general knowledge of:

- PostgreSQL
- SQL
- Linux

Structure of this document

This document is structured as follows:

Chapter 1 Overview of Database Multiplexing Mode

Provides an overview of database multiplexing mode.

Chapter 2 Setting Up Database Multiplexing Mode

Describes how to set up database multiplexing mode.

Chapter 3 Operations in Database Multiplexing Mode

Explains periodic database multiplexing mode.

Chapter 4 Action Required when an Error Occurs in Database Multiplexing Mode

Explains the action required when an error occurs during a database multiplexing mode.

Chapter 5 Managing Mirroring Controller Using WebAdmin

Explains how to set up and manage Mirroring Controller in a streaming replication cluster using WebAdmin.

Appendix A Parameters

Explains the configuration files and parameters required for database multiplexing mode.

Appendix B Supplementary Information on Building the Primary Server and Standby Server on the Same Server

Explains supplementary information on building the primary server and standby server on the same server.

Appendix C User Commands

Explains the user commands.

Appendix D Notes on Performing Automatic Degradation Immediately after a Heartbeat Abnormality

Provides notes when performing automatic degradation unconditionally after a heartbeat abnormality is detected during heartbeat monitoring of an operating system or server.

Appendix E Supplementary Procedure on Configuring for Operation in Database Multiplexing Mode

Explains supplementary procedure on the configuration required for operation in database multiplexing mode.

Appendix F WebAdmin Disallow User Inputs Containing Hazardous Characters

Explains characters not allowed in WebAdmin.

Appendix G Collecting Failure Investigation Data

Explains how to collect data for initial investigation.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

```
Edition 1.0: April 2023
```

Copyright

Copyright 2016-2023 Fujitsu Limited

Contents

Chapter 1 Overview of Database Multiplexing Mode 1
1.1 What is Database Multiplexing Mode 1
1.1.1 Monitoring Using Database Multiplexing Mode 4
1.1.2 Referencing on the Standby Server 6
1.1.2.1 If Prioritizing the Main Job on the Primary Server 6
1.1.2.2 If Performing the Referencing Job on the Synchronous Standby Server 6
1.2 System Configuration for Database Multiplexing Mode 6
1.2.1 Mirroring Controller Resources 9
1.2.1.1 Database Server Resources 9
1.2.1.2 Arbitration Server Resources 10
1.2.2 Mirroring Controller Processes 10
1.2.2.1 Database Server Processes 10
1.2.2.2 Arbitration Server Process 10
1.2.3 Redundancy of the Admin and Log Transfer Networks 11
1.2.4 Notes on CPU Architecture and Products 11
1.3 Deciding on Operation when a Heartbeat Abnormality is Detected 11
1.4 Security in Database Multiplexing 12
1.4.1 Authentication of the Standby Server 14
1.4.2 Encryption of Transaction Logs Transferred to the Standby Server 14

Chapter 2 Setting Up Database Multiplexing Mode 15
2.1 Installation 16
2.2 Preparing for Setup 17
2.2.1 Preparing the Database Server 17
2.2.1.1 Preparing the Backup Disk 17
2.2.2 Preparing the Arbitration Server 17
2.2.2.1 Preparing to Output Error Logs to the Event Log (Windows) 17
2.2.2.2 Security Policy Settings (Windows) 18
2.3 Setting Up the Arbitration Server 18
2.3.1 Configuring the Arbitration Server 18
2.3.2 Creating a User Exit for the Arbitration Server 23
2.3.3 Starting the Mirroring Controller Arbitration Process 24
2.4 Setting Up the Primary Server 24
2.4.1 Setting Up Database Multiplexing Mode on the Primary Server 24
2.4.2 Creating, Setting, and Registering the Primary Server Instance 28
2.4.3 Starting Mirroring Controller on the Primary Server 32
2.5 Setting Up the Standby Server 33
2.5.1 Setting Up Database Multiplexing Mode on the Standby Server 33
2.5.2 Creating, Setting, and Registering the Standby Server Instance 34
2.5.3 Starting Mirroring Controller on the Standby Server 35
2.6 Creating a User Exit for a Database Server 36
2.7 Confirming the Streaming Replication Status 37
2.8 Checking the Connection Status 38
2.8.1 Checking the Connection Status on a Database Server 38
2.8.2 Checking the Connection Status on the Arbitration Server 39
2.9 Creating Applications 40
2.9.1 Application Connection Server Settings 40
2.10 Checking the Behavior 40
2.11 Tuning 40
2.11.1 Tuning to Stabilize the Database Multiplexing Mode 40
2.11.2 Tuning to Stabilize Queries on the Standby Server 40
2.11.3 Tuning to Stabilize Queries on the Standby Server (when Performing Frequent Updates on the Primary Server) 41
2.11.4 Tuning for Optimization of Degradation Using Abnormality Monitoring 41
2.11.4.1 Tuning for Abnormality Monitoring of the Operating System or Server 41
2.11.4.1.1 Tuning Abnormality Monitoring for Operations that Use an Arbitration Server for Automatic Degeneration 42

2.11.4.1.2 Tuning Abnormality Monitoring for Operations that Perform Automatic Degeneration by Calling a User Exit that Determines Degeneration 47
2.11.4.1.3 Tuning Abnormality Monitoring for Operations that Notify Messages 49
2.11.4.1.4 Tuning Abnormality Monitoring for Operations that Perform Automatic Degenerate Unconditionally due to Heartbeat Abnormality 49
2.11.4.2 Tuning for Abnormality Monitoring of Darabase Processes 50
2.11.4.3 Tuning for Abnormality Monitoring of Streaming Replication 51
2.11.4.4 Tuning for Disk Abnormality Monitoring 53
2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances 54
2.13 Setting Automatic Start and Stop of the Mirroring Controller Arbitration Process 56
2.14 Backup Operation 58
2.14.1 Backing up Database Multiplexing Mode Information 58
2.14.2 Database Backup Operation 58

Chapter 3 Operations in Database Multiplexing Mode 60
3.1 Starting and Stopping the Mirroring Controller Arbitration Process 60
3.1.1 Starting the Mirroring Controller Arbitration Process 60
3.1.2 Stopping the Mirroring Controller Arbitration Process 60
3.2 Starting and Stopping Mirroring Controller 61
3.3 Checking the Database Multiplexing Mode Status 63
3.3.1 Checking the Status of the Database Server 63
3.3.2 Checking the Status of the Arbitration Server 64
3.4 Manually Switching the Primary Server 65
3.5 Manually Disconnecting the Standby Server 65
3.6 Action Required when a Heartbeat Abnormality is Detected 66
3.7 Monitoring Mirroring Controller Messages 66
3.8 Server Maintenance 68
3.8.1 Rolling Updates 69
3.8.2 Stopping for Maintenance 74
3.8.3 Arbitration Server Maintenance 74
3.9 Changes in Operation 75
3.9.1 Changes Required when the Standby Server is Stopped 75
3.9.2 Changing from Single Server Mode to Database Multiplexing Mode 76
3.9.3 Changing from Database Multiplexing Mode to Single Server Mode 77
3.9.4 Changing to Database Multiplexing Mode when the Arbitration Server is Used for Automatic Degradation 80
3.9.5 Changing Parameters 81
3.9.6 Uninstalling in Database Multiplexing Mode 81

Chapter 4 Action Required when an Error Occurs in Database Multiplexing Mode 82
4.1 Action Required when Server Degradation Occurs 82
4.1.1 Operations when the Server has Started Degrading after a Switch has Occurred 82
4.1.1.1 Identify Cause of Error and Restore the Standby Server 84
4.1.1.1.1 Stop Mirroring Controller 84
4.1.1.1.2 Recovery of the Mirroring Controller management directory 85
4.1.1.1.3 Identify cause of error and perform recovery 85
4.1.1.2 Rebuild the Standby Server 87
4.1.1.3 Failback of the Primary Server 87
4.1.2 Operations when the Server has Started Degrading after a Disconnection has Occurred 88
4.1.2.1 Identify Cause of Error and Restore the Standby Server 89
4.1.2.1.1 Stop Mirroring Controller 89
4.1.2.1.2 Recovery of the Mirroring Controller management directory 90
4.1.2.1.3 Identify cause of error and perform recovery 90
4.1.2.2 Rebuild the Standby Server 90
4.1.3 Addressing Errors During Degrading Operation 90
4.2 Action Required when Automatic Switch Fails 91
4.3 Action Required when Automatic Disconnection Fails 92
4.4 Action Required when All Database Servers or Instances Stopped 92
4.5 Recovering from an Incorrect User Operation 96

Chapter 5 Managing Mirroring Controller Using WebAdmin 98
5.1 Mirroring Controller Setup 99
5.2 Edit Mirroring Controller Setup 101
5.3 Mirroring Controller Configuration 102
5.4 Stopping Mirroring Controller 103
5.5 Starting Mirroring Controller 104
5.6 Disabling Failover Mode 104
5.7 Enabling Failover Mode 104
5.8 Deleting Mirroring Controller Setup 104
5.9 Status Update after Failover 105
5.10 Action Required when an Error Occurs in the Combined Admin Network and Log Transfer Network 105
5.11 Performing Automatic Degradation Using the Arbitration Server 106

Appendix A Parameters 108
A.1 Parameters Set on the Primary Server 108
A.2 Parameters Set on the Standby Server 110
A.3 Network Configuration File 112
A.4 Server Configuration File 115
A.4.1 Server Configuration File for the Database Servers 115
A.4.2 Arbitration Configuration File 123

Appendix B Supplementary Information on Building the Primary Server and Standby Server on the Same Server 128
B.1 Backup Data Storage Destination Directory 128
B.2 How to Execute the mc_ctl Command 128

Appendix C User Commands 130
C.1 Fencing Command 130
C.2 Arbitration Command 131
C.3 State Transition Commands 132
C.3.1 Post-switch Command 132
C.3.2 Pre-detach Command 133
C.3.3 Post-attach Command 133

Appendix D Notes on Performing Automatic Degradation Immediately after a Heartbeat Abnormality 135

Appendix E Supplementary Procedure on Configuring for Operation in Database Multiplexing Mode 138
E.1 Security Policy Settings 138
E.2 Windows Firewall Settings 138

Appendix F WebAdmin Disallow User Inputs Containing Hazardous Characters 140

Appendix G Collecting Failure Investigation Data 141

Index 143

Chapter 1 Overview of Database Multiplexing Mode

This chapter provides an overview of database multiplexing mode.

Point

In this and subsequent chapters, the word "Mirroring Controller" may be used in the process or management directory name or explanation.

1.1 What is Database Multiplexing Mode

Database multiplexing mode is an operation mode (log shipping mode) based on PostgreSQL streaming replication. Other software such as cluster software is not required.

This mode replicates the database on all servers that comprise the cluster system. It achieves this by transferring the updated transaction logs of the database from the server that receives the updates (primary server) to another server (standby server), and then reflecting them on the standby server. The client driver automatically distinguishes between the primary and standby servers, so applications can be connected transparently regardless of the physical server.

It consists of a feature that detects faults in the elements that are essential for the continuity of the database operation (such as the database process, disk, and network), as well as simplified switchover and standby server disconnection features. Furthermore, referencing can be performed on the standby server. The database will be copied in synchronous mode.

Note

If using WebAdmin or Mirroring Controller, Fujitsu Enterprise Postgres supports cluster systems comprising one primary server and one standby server.

- Although it is possible to connect an asynchronous standby server to the cluster system as an additional server, the standby server is not targeted for monitoring by Mirroring Controller.
- A synchronous standby server cannot be connected to the cluster system as an additional server.

See

The streaming replication feature is not described in this manual.

Refer to "High Availability, Load Balancing, and Replication" in the PostgreSQL Documentation for information on the streaming replication feature.

Figure 1.1 Failover from the primary server to the standby server

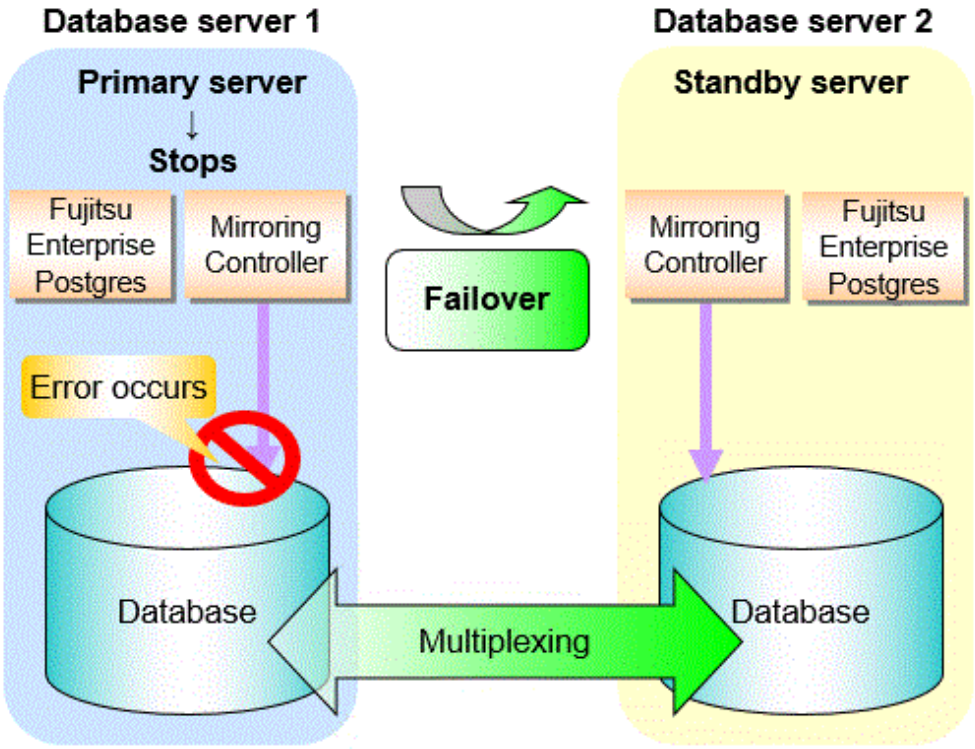
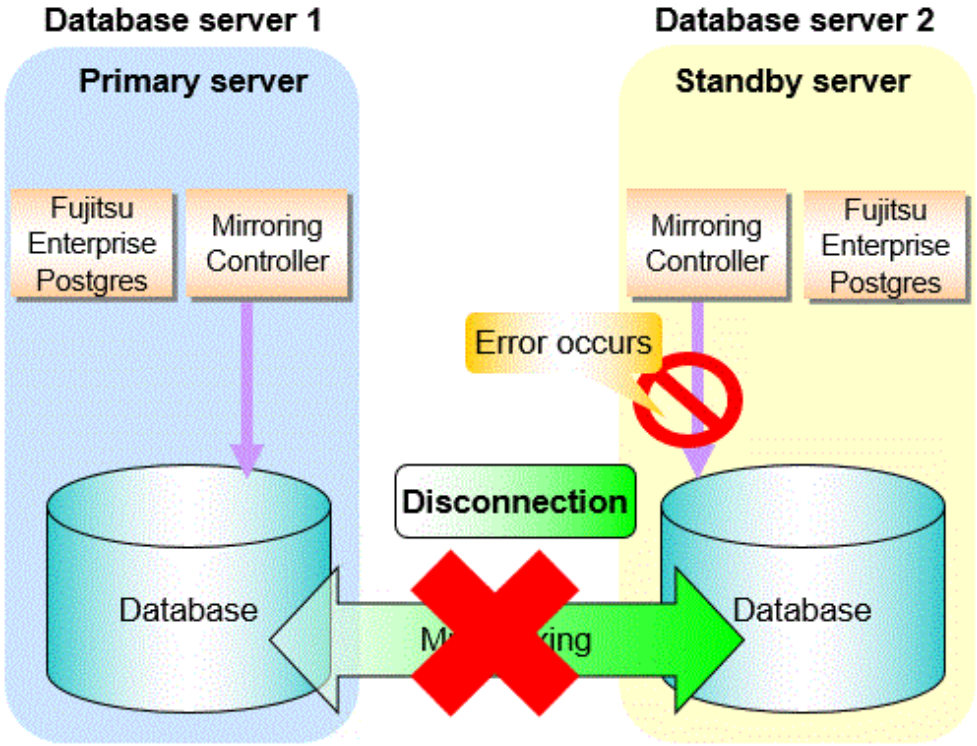


Figure 1.2 Standby server disconnection



Database degradation using the arbitration server

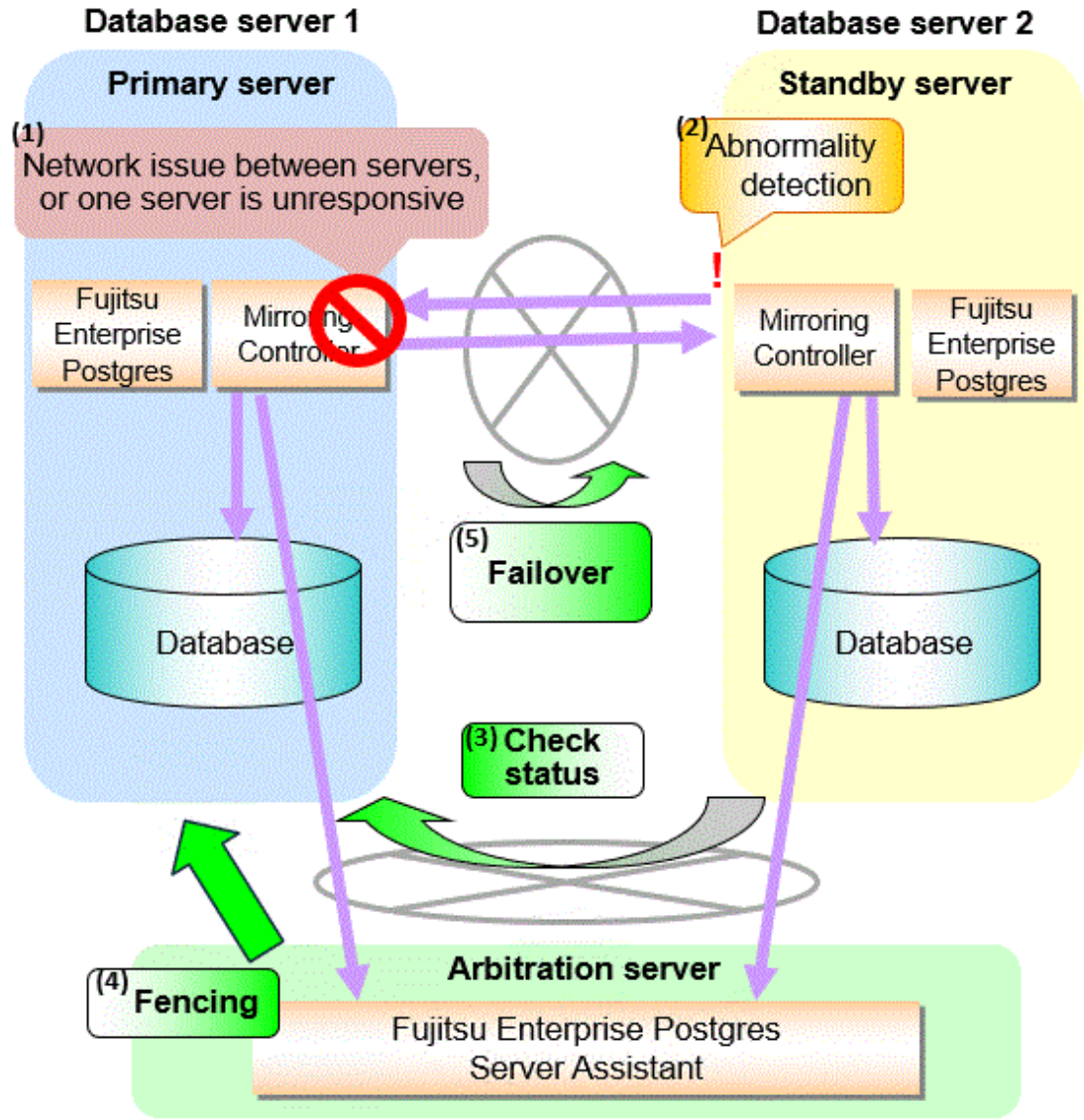
Fujitsu Enterprise Postgres provides the Server Assistant that objectively determines the status of database servers as a third party, and if necessary, isolates affected databases if the database servers are unable to accurately ascertain their mutual statuses in database multiplexing

mode, such as due to a network error between database servers, or server instability. Database degradation can be performed by using the server (arbitration server) on which the Server Assistant is installed.

For database degradation using the arbitration server, if the database servers are unable to check their mutual statuses (due to a network error between database servers or server instability), then the database server queries the arbitration server for the status of the other database server. If it is determined based on the heartbeat result that the status is unstable, the applicable database server will be isolated from the cluster system (fencing). The arbitration server periodically heartbeats the database server so that it can respond immediately to queries from the database server. The fencing process can be customized according to the environment where Mirroring Controller is used.

Additionally, the database servers are always performing their heartbeats for the arbitration server so that it can perform check requests any time.

Figure 1.3 Database degradation using the arbitration server



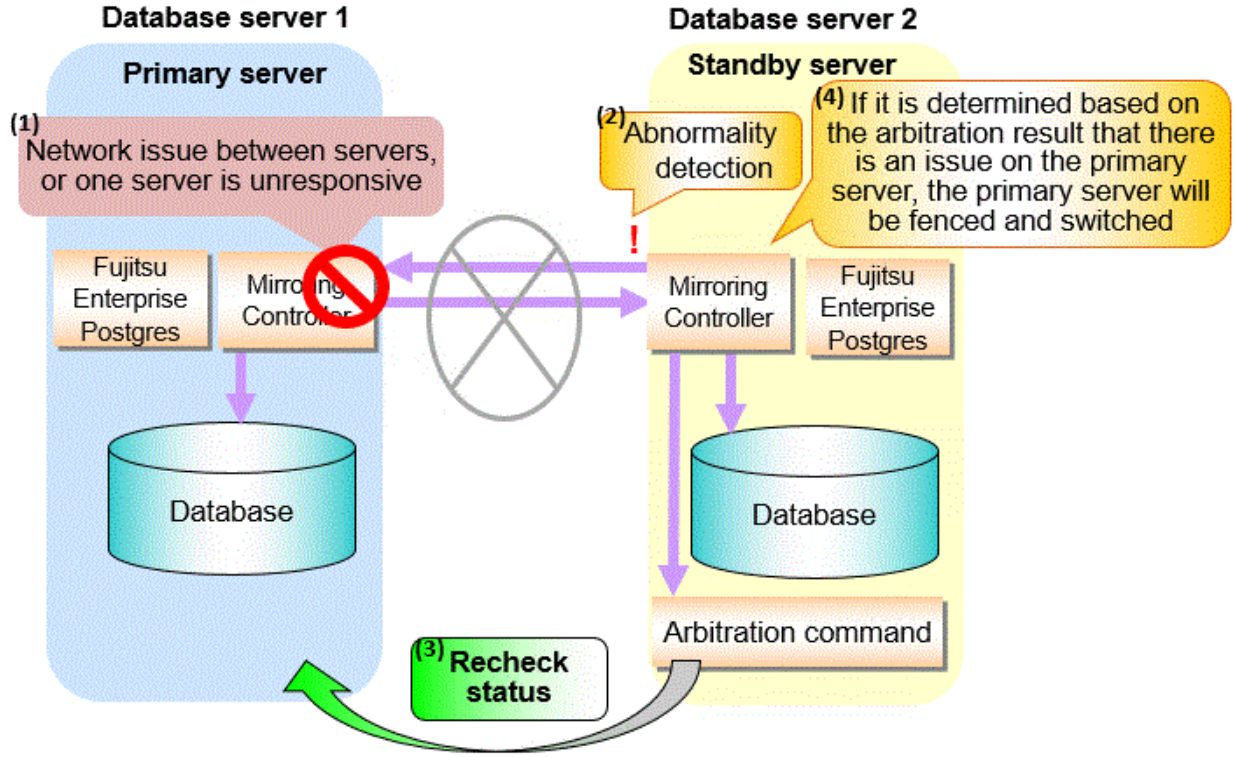
Note

Install the arbitration server on a different physical server to that of the database server. Refer to "1.2 System Configuration for Database Multiplexing Mode" for information on the system configuration when using the arbitration server.

Database degradation using the arbitration command

The arbitration command is a user command that performs arbitration processing in lieu of the arbitration server. If an arbitration server cannot be deployed, arbitration of the database server can be performed using the arbitration command.

Figure 1.4 Database degradation using the arbitration command



See

Refer to "2.6 Creating a User Exit for a Database Server" or "Appendix C User Commands" for information on user commands.

1.1.1 Monitoring Using Database Multiplexing Mode

In database multiplexing mode, perform the monitoring below.

- Operating system or server failures, and no-response state

  By generating a heartbeat between Mirroring Controller on each server, operating system or server errors are detected and acknowledged between the relevant servers.

  The optimal operating method for environments where database multiplexing mode is performed can be selected from the following:

  - Use the arbitration server to perform automatic degradation (switch/disconnect)

    This is the default method.

    The arbitration server objectively determines the status of database servers, then isolates and degrades from the cluster system the ones with an unstable status.

    Refer to "Database degradation using the arbitration server" for details.

  - Call the user exit (user command) that will perform the degradation decision, and perform automatic degradation

    If the arbitration server cannot be installed, select if arbitration processing can be performed by the user instead.

    Mirroring Controller queries the user exit on whether to degrade. The user exit determines the status of the database server, and notifies Mirroring Controller whether to perform degradation.

Refer to "Database degradation using the arbitration command" for details.

- Notification messages

  Use this method if using a two-database server configuration.

  Mirroring Controller outputs messages to the system log when an abnormality is detected. This ensures that a split brain will not occur due to a heartbeat abnormality - however, automatic switching will not be performed if the primary server operating system or server fails or becomes unresponsive.

- Perform automatic degradation unconditionally after a heartbeat abnormality

  This method is handled as in FUJITSU Enterprise Postgres 9.6 or earlier versions.

  This method is not recommended, because Mirroring Controller unconditionally will perform automatic degradation after heartbeat abnormalities.

- Database process failures, and no-response state

  Mirroring Controller periodically accesses the database processes and checks the status. A process error is detected by monitoring whether an access timeout occurs.

- Disk failure

  Mirroring Controller periodically creates files on the data storage destination disk below. A disk error is detected when an I/O error occurs.

  - Data storage destination disk
  - Transaction log storage destination disk
  - Tablespace storage destination disk

  Failures that can be detected are those that physically affect the entire system, such as disk header or device power failures.

- Streaming replication issue

  Mirroring Controller detects streaming replication issues (log transfer network and WAL send/receive processes) by periodically accessing the PostgreSQL system views.

- Mirroring Controller process failure and no response

  In order to continue the monitoring process on Mirroring Controller, Mirroring Controller process failures and no responses are also monitored.

  The Mirroring Controller monitoring process detects Mirroring Controller process failures and no responses by periodically querying the Mirroring Controller process. If an issue is detected, Mirroring Controller is automatically restarted by the Mirroring Controller monitoring process.

Point

- If output of messages is selected as the operation to be performed when a heartbeat abnormality is detected during heartbeat monitoring of the operating system or server, automatic degradation will not be performed.
  However, if an issue in the WAL send process is detected on the primary server, then the standby server will be disconnected, and as a result an automatic disconnection may be performed even if the standby server operating system or server fails or becomes unresponsive.
- You can select in the parameters if the primary server will be switched if a database process is unresponsive or if tablespace storage destination disk failure is detected. Refer to "Appendix A Parameters" for details.
- If the standby server was disconnected, Mirroring Controller will automatically comment out the synchronous_standby_names parameter in the postgresql.conf file of the primary server. Accordingly, you can prevent the application processing for the primary server being stopped.

Note

If the role of primary server was switched to another server and then starts degrading, the original primary server will not become the standby server automatically. Remove the cause of the error, and then change the role of the original primary server to the server currently acting as standby server. Refer to "4.1 Action Required when Server Degradation Occurs" for details.

1.1.2 Referencing on the Standby Server

1.1.2.1 If Prioritizing the Main Job on the Primary Server

If a reference job is performed on the standby server and the primary server is switched, this may impact the main job from the point of view of load and conflict. This is because, on the new primary server (that is, the original standby server), both the main job that was being executed on the original primary server and the reference job that was being continued on the original standby server will be processed.

Therefore, to degrade the reference job (so that the impact on the main job is reduced), you can select the user exit below to disconnect the reference job that was performed on the original standby server.

- Post-switch command

Note

If continuing with the referencing job after switching the primary server, give careful consideration to the server resource estimates, and the likely impact on performance.

1.1.2.2 If Performing the Referencing Job on the Synchronous Standby Server

If an issue such as a log transfer network failure obstructs the continuation of a job on the primary server, the standby server may be automatically disconnected from the cluster system.

Therefore, if operating the reference job on the assumption that the connection destination is the synchronous standby server, you can select to temporarily stop the job by using the user exit or the feature below, so that unexpected referencing of past data does not occur as a result of the disconnection.

- Pre-detach command
- Forced stoppage of the standby server instance on disconnection (specify in the parameter of the server configuration file)

Additionally, if the standby server is incorporated into the cluster system, reference jobs can be started or resumed by using the user exit below.

- Post-attach command

See

- Refer to "2.6 Creating a User Exit for a Database Server" or "Appendix C User Commands" for information on each user exit.
- Refer to "A.4.1 Server Configuration File for the Database Servers" for information on the server configuration file of the database server.

Point

Mirroring Controller will continue processing regardless of the processing result of the above user exits and features.

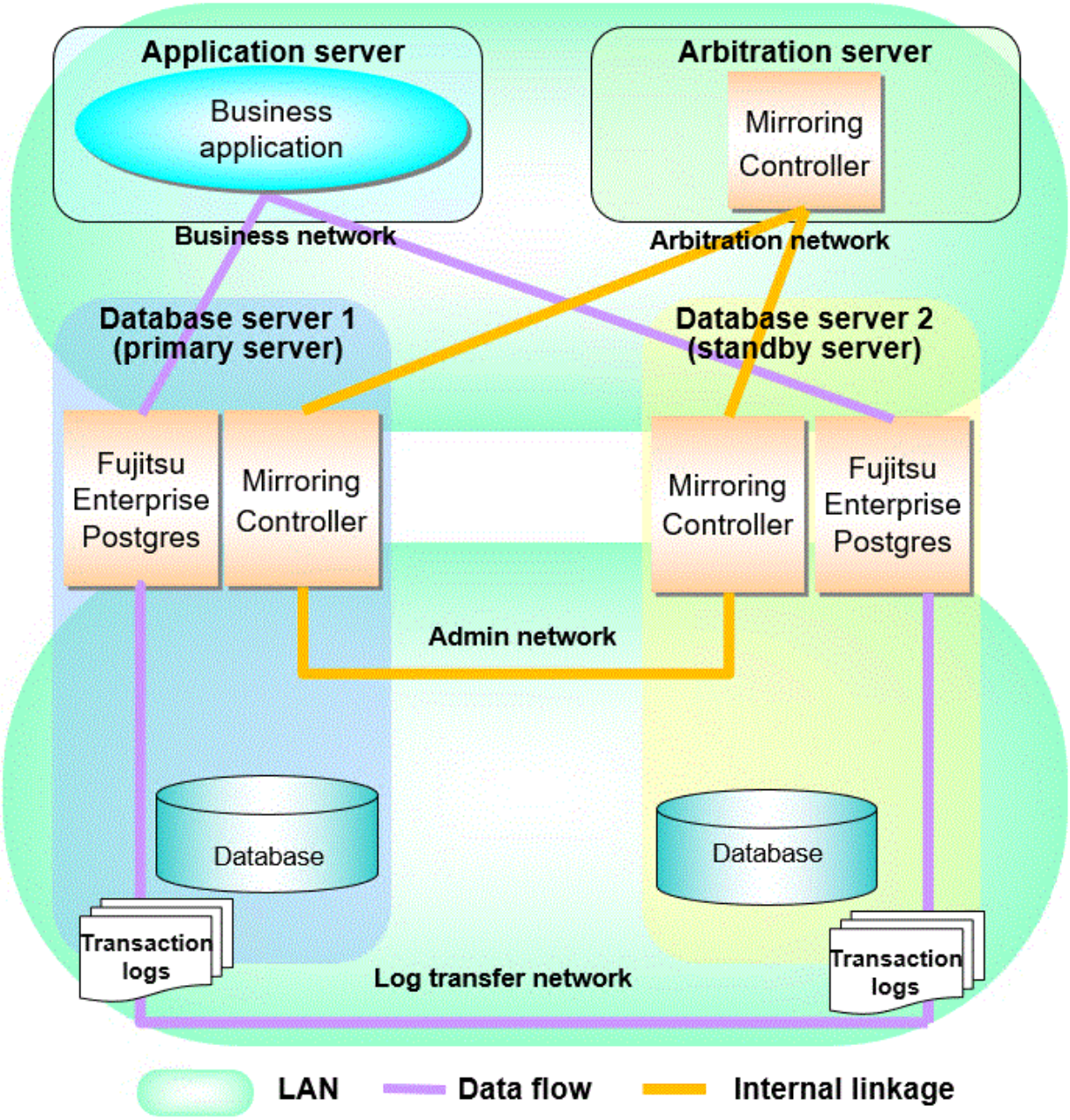
1.2 System Configuration for Database Multiplexing Mode

This section explains the products, features, and networks that are part of a database multiplexing system.

The following table shows the network types uses by database multiplexing systems.

Network type	Description
Job network	Network between the application that accesses the database, and the database server.
Arbitration network	Network used by the arbitration server to check the status of the primary server and standby server, and communicate with Mirroring Controller of the database servers. Additionally, if the job network is disconnected from outside, it can also be used as the arbitration network. Refer to "1.4 Security in Database Multiplexing" for details on network security.
Admin network	Network used by the primary server and the standby server to monitor each other using Mirroring Controller, and to control Mirroring Controller of other servers.
Log transfer network	Network used to transfer the transaction logs of the database, which is part of database multiplexing.

Figure 1.5 System configuration for database multiplexing mode



The arbitration server is installed to check the database server status as a third party, and to perform fencing. Therefore, to obtain the intended benefits, consider the following.

- Install the arbitration server on a different server to that of the database server.
- For the arbitration network, use a network that will not be impacted by line faults or the load on the admin network or log transfer network. This is necessary to correctly determine issues on the admin network or log transfer network.

Point

- The arbitration server can also be used as an application server. However, consider the server load.
- It is recommended to link the arbitration server with other cluster systems, in order to provide redundancy.
- Use the arbitration server in combination with the same version of Fujitsu Enterprise Postgres as that of the primary server and standby server.

- The arbitration server can be built on a different platform to that of the database server.

Note

Because the ping command of the operating system is used for heartbeat monitoring of the database server, configure the network so that ICMP can be used on the admin network and the arbitration network.

1.2.1 Mirroring Controller Resources

This section describes the database server and arbitration server resources of Mirroring Controller.

1.2.1.1 Database Server Resources

The only Mirroring Controller resource is the Mirroring Controller management directory, which stores the files that define the Mirroring Controller behavior, and the temporary files that are created when Mirroring Controller is active.

Note

- Do not create the Mirroring Controller management directory in a directory managed by Fujitsu Enterprise Postgres, otherwise it may be deleted by mistake or may cause unexpected problems when Fujitsu Enterprise Postgres recovery is performed (such as old version of files being restored).

  Refer to "Preparing Directories for Resource Deployment" in the Installation and Setup Guide for Server for information on the directories managed by Fujitsu Enterprise Postgres.

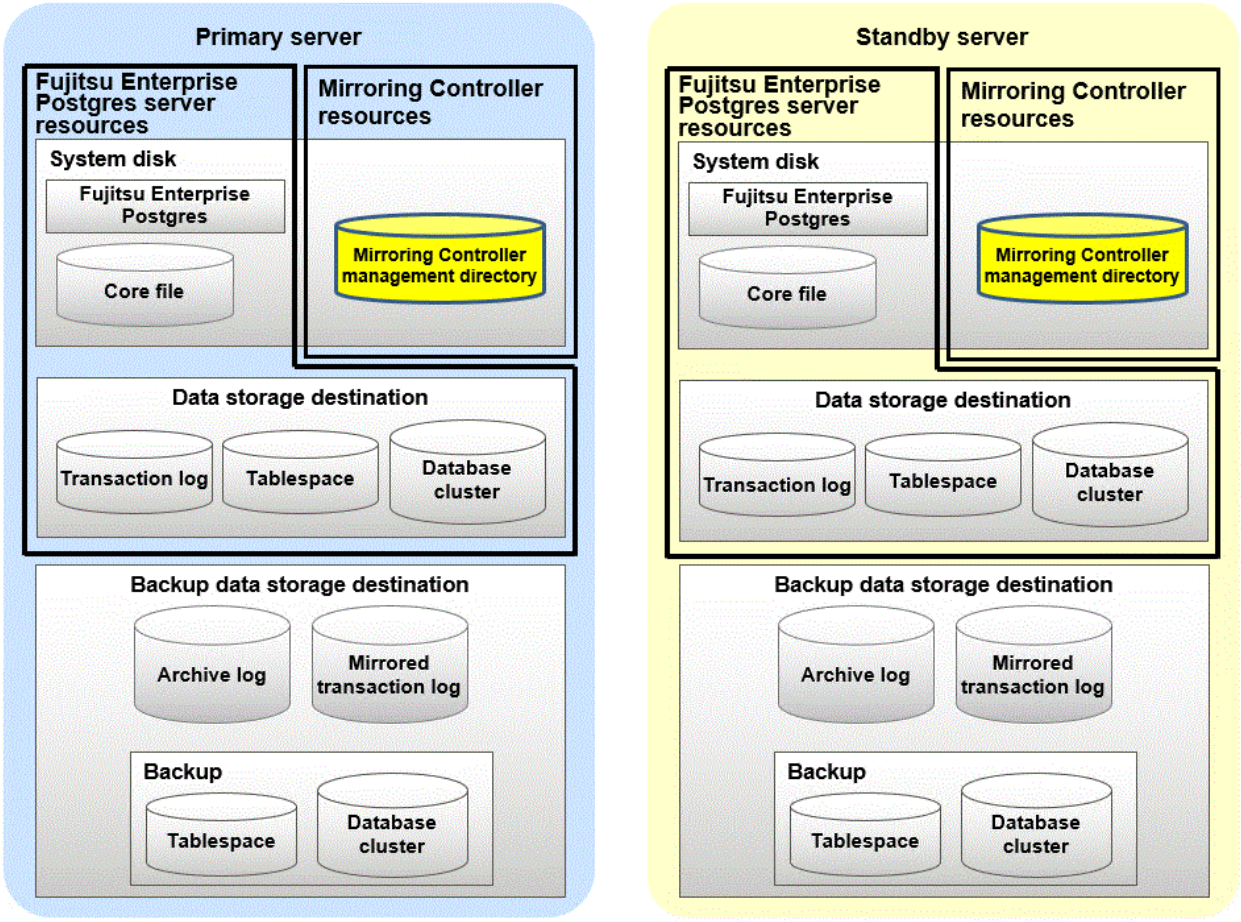
- The backup methods described in "Backing Up the Database" in the Operation Guide cannot be used to back up the Mirroring Controller resources. Therefore, users must obtain their own backup of Mirroring Controller resources, in addition to Fujitsu Enterprise Postgres server resources. Retrieve backups after stopping Mirroring Controller.

- If the automatic switch/disconnection is enabled, do not edit synchronous_standby_names for the Mirroring Controller monitoring target instance. Otherwise, if Mirroring Controller is switched after editing, data may be lost or SQL access may stop.

- If you are building on a virtual machine or cloud, make sure the virtual machines are on different physical servers. Refer to your virtual machine software and cloud vendor documentation for instructions on how to deploy virtual machines.

The content on the primary server will be backed up. You cannot tell which server is the primary server to be backed up, because switching and failback may be performed between the servers. It is also impossible to tell which server is to be restored using the backed up data. Accordingly, ensure that you create a backup of each server when it is working as the primary server.

Figure 1.6 Configuration when backing up Mirroring Controller resources



1.2.1.2 Arbitration Server Resources

The only arbitration server resource is the Mirroring Controller arbitration process management directory. This directory stores the files that define the Mirroring Controller arbitration process behavior and the temporary files created when Mirroring Controller is active.

1.2.2 Mirroring Controller Processes

This section describes the database server and arbitration server processes of Mirroring Controller.

1.2.2.1 Database Server Processes

The database server processes comprise the Mirroring Controller process and Mirroring Controller monitoring process.

Process type	Description
Mirroring Controller process	Performs operating system/server and process heartbeat monitoring and disk abnormality monitoring between database servers. Additionally, it issues arbitration requests to the arbitration server.
Mirroring Controller monitoring process	Performs heartbeat monitoring of the Mirroring Controller process. If the Mirroring Controller process returns no response or is down, the monitoring process is restarted automatically.

1.2.2.2 Arbitration Server Process

The only arbitration process is the Mirroring Controller arbitration process.

Process type	Description
Mirroring Controller arbitration process	Performs rechecks for issues detected on the primary server or the standby server. Additionally, this process performs fencing if it determines that there is an issue on the primary server or the standby server.

1.2.3 Redundancy of the Admin and Log Transfer Networks

The admin network is an important one, because it is used by Mirroring Controller to check the status of each server.

Additionally, the log transfer network is an important one, because it is necessary to ensure data freshness.

Accordingly, configure a failure-resistant network by implementing network redundancy via channel bonding provided by the operating system or network driver vendor.

1.2.4 Notes on CPU Architecture and Products

Use the same CPU architecture (endian) for the primary server, standby server, and the arbitration server.

A server using only PostgreSQL streaming replication cannot be specified as the database multiplexing system log transfer destination.

1.3 Deciding on Operation when a Heartbeat Abnormality is Detected

The operation to be performed when a heartbeat abnormality is detected using operating system/server heartbeat monitoring is decided on according to the environment where database multiplexing mode is performed or the operating method.

It is possible to select from the four operations below, and specify this in the parameters of Mirroring Controller:

- Use the arbitration server to perform automatic degradation (switch/disconnect)
- Call the user exit (user command) that will perform the degradation decision, and perform automatic degradation
- Notification messages
- Perform automatic degradation unconditionally (switch/disconnect)

The table below shows if jobs can be continued on the primary server when an issue is detected during heartbeat monitoring of the operating system/server.

Continuation of jobs on the primary server when an issue is detected during heartbeat monitoring of the operating system/server

Operation	Abnormal event				
	Server/operating system failures or no responses		Admin network issue	Log transfer network issue	Issue on a network for both admin and log transfer
	Primary server	Standby server			
Automatic degradation using the arbitration server	Y (switch)	Y (disconnect)	Y	Y (disconnect)	Y (disconnect)
Call a user exit and perform automatic degradation	Y (switch)	Y (disconnect)	Y	Y (disconnect)	Y (disconnect)
Notification messages	N (message notification only)	N (message notification only)	Y	Y (disconnect)	Y (disconnect)
Unconditional automatic degradation	Y (switch)	Y (disconnect)	N (split brain occurs)	Y (disconnect)	N (split brain occurs)

Y: Job can be continued

N: Job cannot be continued

1.4 Security in Database Multiplexing

The database server replicates the database on all servers that comprise the cluster system. It achieves this by transferring and reflecting the updated transaction logs of the database from the primary server to the standby server.

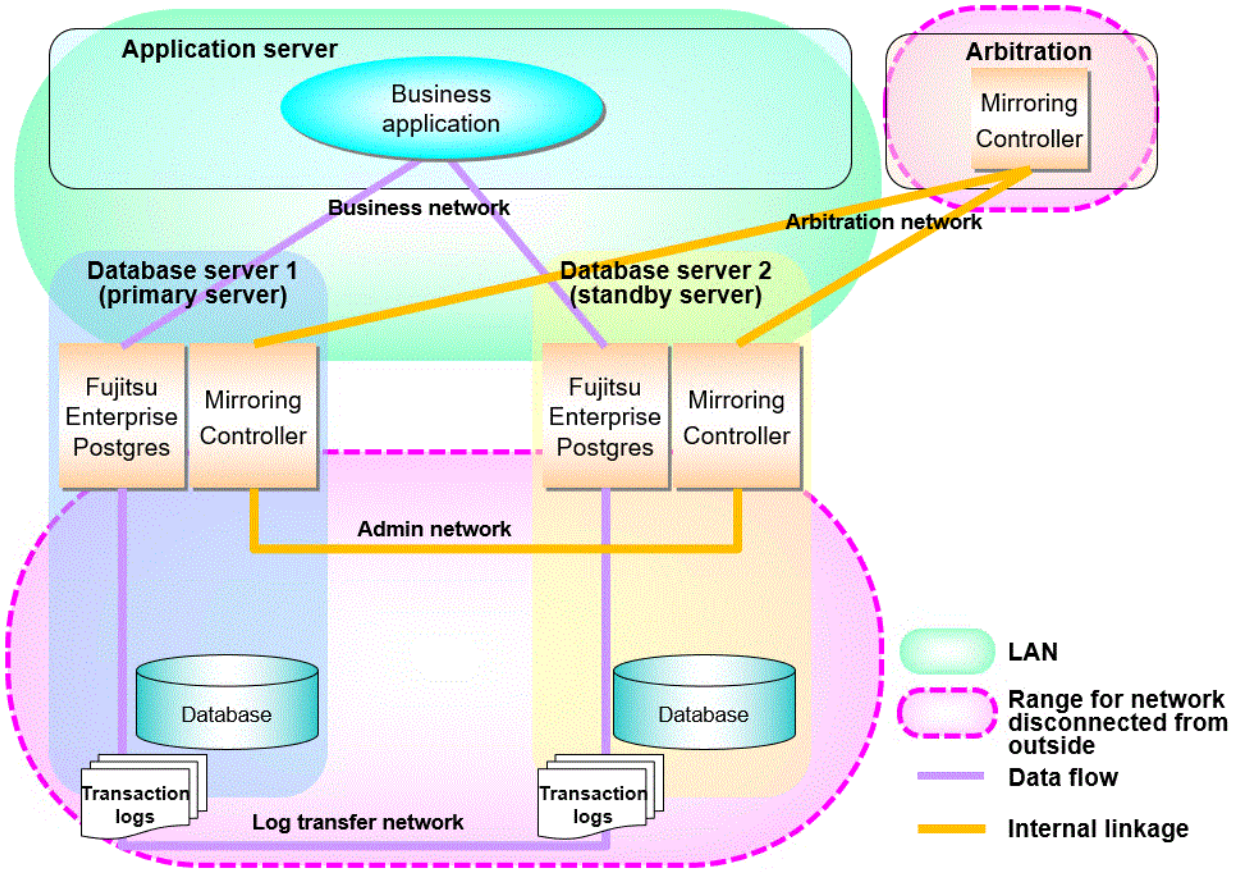
To safeguard the database against unauthorized access and preserve data confidentiality in transaction log transfers, carefully consider security and take note of the following when performing database multiplexing:

- Do not use trust authentication when using replication connection.
- Configure the admin network and the log transfer network so that they cannot be connected from the outside, as shown in Figure 1.7 Security.

Additionally, for the line on which Mirroring Controller connects from the database server to the arbitration server, take note of the following points and consider security carefully.

- Build a network with the arbitration server disconnected from outside, as shown in Figure 1.7 Security.

Figure 1.7 Security



However, it may not always be possible to adopt the configuration mentioned above. For example, you may want to place the servers in a nearby/neighboring office to minimize network delays.

In this case, combine the following features to enhance security:

- Authentication of the Standby Server
- Encryption of Transaction Logs Transferred to the Standby Server

When these features are combined, security will be achieved as shown below.

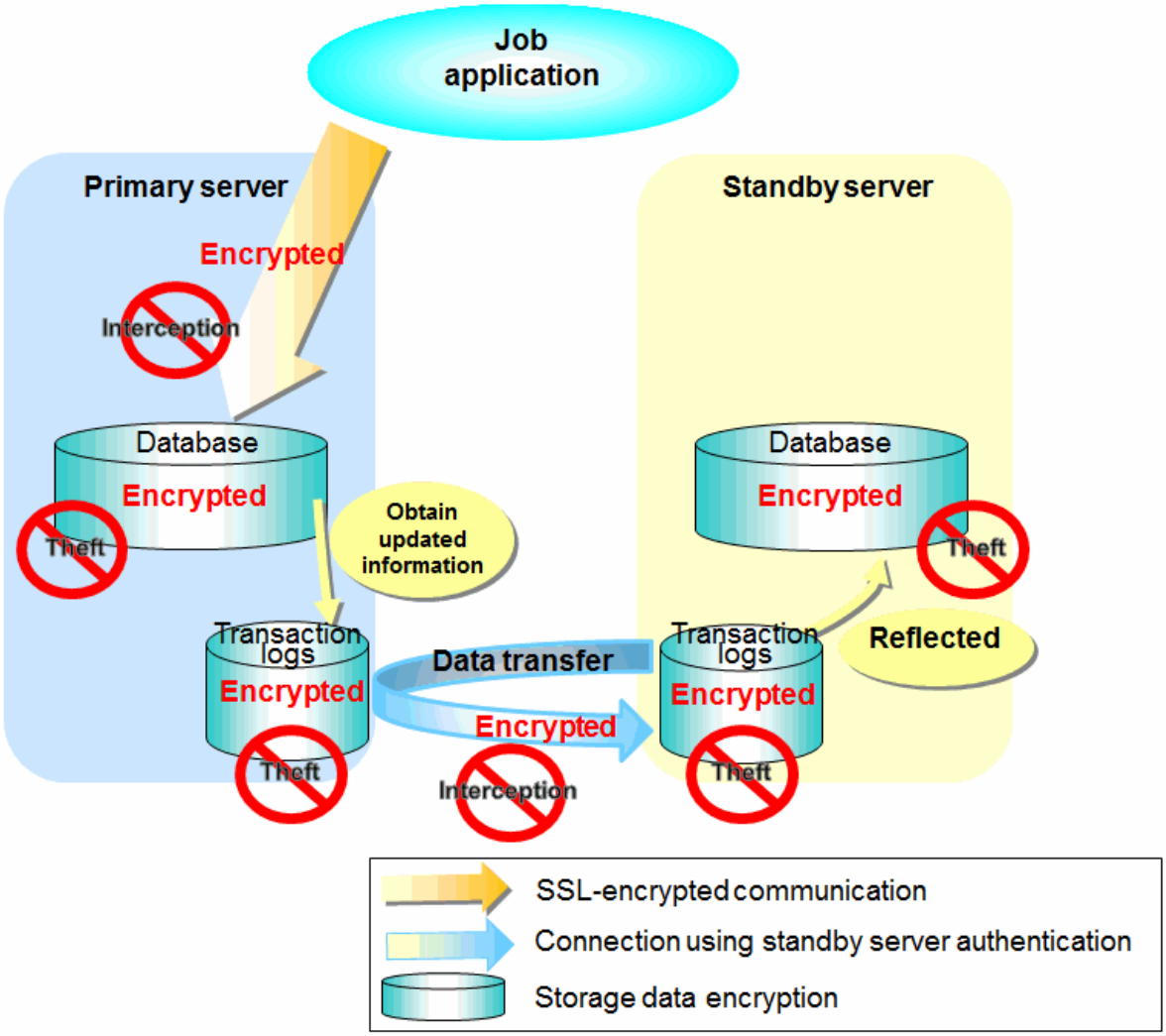
Point

If the job network is disconnected from outside, it can be used as the arbitration network. However, if a network is to be used as both a job network and arbitration network, consider the load on the network.

Note

If a port is blocked (access permission has not been granted) by a firewall, etc., enable use of the target port by granting access. Refer to the vendor document for information on how to open (grant access permission to) a port. Consider the security risks carefully when opening ports.

Figure 1.8 Security achieved when standby server authentication is combined with transaction log encryption



See

Refer to "Performing Database Multiplexing" under "Configuring Secure Communication Using Secure Sockets Layer" in the Operation Guide for information on encrypting SSL communications.

1.4.1 Authentication of the Standby Server

You can prevent spoofing connections from an external server purporting to be the standby server by using authentication with a user name and password.

Configure the setting in the primary server pg_hba.conf file so that authentication is performed for connections from the standby server in the same way as for connections from the client.

Refer to "Client Authentication" in the PostgreSQL Documentation for information on content that can be configured in pg_hba.conf.

1.4.2 Encryption of Transaction Logs Transferred to the Standby Server

In case the authentication of the standby server is breached so that a malicious user purporting to be the standby server can spoof data, the transaction log data can be encrypted to prevent it from being deciphered. The transparent data encryption feature is used to encrypt the data.

Refer to "Protecting Storage Data Using Transparent Data Encryption" in the Operation Guide for details.

Chapter 2 Setting Up Database Multiplexing Mode

This chapter describes how to set up database multiplexing mode, and how to check it.

Users who perform setup and operations on the database server

Setup and operations of the database server must be performed by the instance administrator user.

Users who perform setup and operations on the arbitration server

The following users may perform setup and operations on the arbitration server when it is used for automatic degradation.

Linux

Any operating system user.

Windows

Any user with administrator privileges. This user must be assigned the "Log on as a service" user right.

Point

- Mirroring Controller selects a database superuser as the user who will connect to the database instance. This enables instance administrator users and database superusers who operate the Mirroring Controller commands to run database multiplexing mode in different environments.
- The application name for connecting to the database instance is "mc_agent".

Matching the system times

Before starting the setup, ensure that the times in the primary server, standby server and arbitration server match, by using the operating system time synchronization feature, for example.

The tolerated difference is approximately one second.

If the system times are not synchronized (because the tolerated difference is exceeded, for example), problem investigation may be affected.

Configuring ICMP

Because the ping command of the operating system is used for heartbeat monitoring of the database server, configure the network so that ICMP can be used on the admin network and the arbitration network. Refer to the relevant operating system procedure for details.

Setup

The setup procedure is shown in the table below. However, the procedure on the arbitration server should be performed only when the arbitration server is used for automatic degradation. A distinction is made between the procedures on the primary server and standby server according to whether the arbitration server is used.

Step	Task			Refer to
	Primary server	Standby server	Arbitration server	
1	Installation			2.1 Installation
2	Preparing the database server		Preparing the arbitration server	2.2 Preparing for Setup
3			Configuring the arbitration server	2.3.1 Configuring the Arbitration Server
4			Creating a user exit	2.3.2 Creating a User Exit for the Arbitration Server

Step	Task			Refer to
	Primary server	Standby server	Arbitration server	
5			Starting the arbitration process	2.3.3 Starting the Mirroring Controller Arbitration Process
6	Setting up database multiplexing mode			2.4.1 Setting Up Database Multiplexing Mode on the Primary Server
7	Creating, setting, and registering the instance			2.4.2 Creating, Setting, and Registering the Primary Server Instance
8	Creating a user exit			2.6 Creating a User Exit for a Database Server
9	Starting Mirroring Controller			2.4.3 Starting Mirroring Controller on the Primary Server
10		Setting up database multiplexing mode		2.5.1 Setting Up Database Multiplexing Mode on the Standby Server
11		Creating, setting, and registering the instance		2.5.2 Creating, Setting, and Registering the Standby Server Instance
12		Creating a user exit		2.6 Creating a User Exit for a Database Server
13		Starting Mirroring Controller		2.5.3 Starting Mirroring Controller on the Standby Server
14	Confirming the streaming replication status			2.7 Confirming the Streaming Replication Status
15	Checking the connection status			2.8.1 Checking the Connection Status on a Database Server
16		Checking the connection status		2.8.1 Checking the Connection Status on a Database Server
17			Checking the connection status	2.8.2 Checking the Connection Status on the Arbitration Server
18	Creating applications			2.9 Creating Applications
19	Checking the behavior			2.10 Checking the Behavior

Explanations for each step are provided below.

Information

- The setup procedure is also the same when changing the mode on a single server to database multiplexing mode. In this case, omit the installation of Fujitsu Enterprise Postgres and the creation of the instance.

  Refer to "3.9.2 Changing from Single Server Mode to Database Multiplexing Mode" for details.

- The primary and standby server can be pseudo-configured on the same server for system testing, for example. In this case, the setup can be performed using the same procedure, however there will be some supplementary steps.

  Before performing the setup, refer to "Appendix B Supplementary Information on Building the Primary Server and Standby Server on the Same Server".

2.1 Installation

Refer to the manuals below, and then install the product.

See

- Refer to the Installation and Setup Guide for Server for details on how to install Fujitsu Enterprise Postgres.
- Refer to the Installation and Setup Guide for Server Assistant for information on installing the Server Assistant on the arbitration server.

Note

Do not use the arbitration server also as a database server. The arbitration server is installed to check the database server status as a third party, and to perform fencing. Using the arbitration server also as a database server nullifies the effectiveness of the arbitration server.

2.2 Preparing for Setup

This section describes the preparation required before setting up Mirroring Controller.

2.2.1 Preparing the Database Server

2.2.1.1 Preparing the Backup Disk

In Mirroring Controller, by performing a backup, recovery is possible even if all server disks are corrupted.

The content on the primary server should be backed up. However, through switching and failback, the standby server may also become the primary server. Accordingly, prepare each of the backup disk devices for the primary and standby servers. Perform backup on the primary server used at the time of the backup.

2.2.2 Preparing the Arbitration Server

2.2.2.1 Preparing to Output Error Logs to the Event Log (Windows)

This section explains the preparatory tasks for outputting error logs to the event log.

Note

If no event source name is registered, messages output to the event log may be incomplete.

Configuring each server

Event logs for the Mirroring Controller commands on the arbitration server may be output with the default event source name "MirroringControllerArbiter". Therefore, register this default event source name beforehand.

```
Example)
The following is an example in which the DLL of a 64-bit product is registered under the default event
source name. Note that "<x>" indicates the product version.
```

```
> regsvr32 "c:\Program Files\Fujitsu\fsepv<x>assistant64\lib\mcarbevent.dll"
```

Setting each instance

You can output messages to any event source named by the user, so that messages output to the event log can be identified by each instance.

```
Example)
The following is an example in which the DLL of a 64-bit product is registered under the event source
name "Mirroring Controller arbiter1". Note that "<x>" indicates the product version.
```

```
> regsvr32 /n /i:"Mirroring Controller arbiter1" "c:\Program Files\Fujitsu\fsepv<x>assistant64\lib
\mcarbevent.dll"
```

The parameter must be edited for each instance. Refer to "A.4.2 Arbitration Configuration File" and set the event_source parameter.

If installing multiple versions

If Fujitsu Enterprise Postgres is already installed on the same machine, search for the key below in Registry Editor, and make a note of the path of the registered DLL. Afterwards, register a new DLL using the default event source name.

Use the DLL path that you made a note of in the above step when re-registering the default event source name during uninstallation.

```
MirroringControllerArbiter
```

2.2.2.2 Security Policy Settings (Windows)

On the arbitration server, operating system user accounts that operate the Mirroring Controller arbitration process must be assigned the "Log on as a service" user right in order to use Windows Services to start and stop the Mirroring Controller arbitration process.

If the security settings to enable this have not been configured, refer to "E.1 Security Policy Settings" and configure the settings.

2.3 Setting Up the Arbitration Server

This section explains how to set up the arbitration server.

2.3.1 Configuring the Arbitration Server

This section explains how to set up database multiplexing mode on the arbitration server.

In database multiplexing mode, the files that are required for operations are managed in the Mirroring Controller arbitration process management directory.

There is one Mirroring Controller arbitration process management directory for each arbitration process.

Point

The arbitration process for each database multiplexing system can be started on a single arbitration server.

See

- Refer to the Reference for information on the mc_arb command.
- Refer to "Appendix A Parameters" for information on the parameters to be edited for the setup.

Perform the following procedure:

Linux

1. On the arbitration server, log in as any operating system user who starts and stops the arbitration process.
2. Configure the environment variables.

   Set the following environment variables:

   - PATH

     Add the installation directory "/bin".

   - MANPATH

     Add the installation directory "/share/man".

Example

The following example configures environment variables when the installation directory is "/opt/fsepv<*x*>assistant".

Note that "<*x*>" indicates the product version.

sh, bash

```
$ PATH=/opt/fsepv<x>assistant/bin:$PATH ; export PATH
$ MANPATH=/opt/fsepv<x>assistant/share/man:$MANPATH ; export MANPATH
```

csh, tcsh

```
$ setenv PATH /opt/fsepv<x>assistant/bin:$PATH
$ setenv MANPATH /opt/fsepv<x>assistant/share/man:$MANPATH
```

3. Create the Mirroring Controller arbitration process management directory that will store the files required by the arbitration server.

   Use ASCII characters in the Mirroring Controller arbitration process management directory.

4. In the network configuration file (network.conf), define the Mirroring Controller network configuration that will be managed by the Mirroring Controller arbitration process.

   Create network.conf in the Mirroring Controller arbitration process management directory, based on the sample file. For network.conf, set read and write permissions only for the operating system user who starts and stops the arbitration process in step 1.

   If users other than this are granted access permissions, the mc_arb command will not work. Accordingly, users other than the operating system user who starts and stops the arbitration process in step 1 are prevented from operating the Mirroring Controller arbitration process.

   Sample file

   ```
   /installDir/share/mcarb_network.conf.sample
   ```

   In network.conf, specify the IP address or host name and port number of the primary server and standby server, and define the Mirroring Controller network configuration that will be managed by the Mirroring Controller arbitration process.

   Refer to "A.3 Network Configuration File" for details.

   A definition example is shown below.

   ```
   Example)
   The IDs of the servers are set to "server1" and "server2", and their port numbers are set to
   "27541".
   ```

   ```
   server1 192.0.3.100 27541
   server2 192.0.3.110 27541
   ```

5. In the arbitration configuration file (arbitration.conf), define the information related to control of the Mirroring Controller arbitration process.

   Create arbitration.conf in the Mirroring Controller arbitration process management directory, based on the sample file. For arbitration.conf, set read and write permissions only for the operating system user who starts and stops the arbitration process in step 1. If users other than this are granted access permissions, the mc_arb command will not work.

   Sample file

   ```
   /installDir/share/mcarb_arbitration.conf.sample
   ```

   Set the parameters shown in the table below in arbitration.conf.

Table 2.1 Parameters

Parameter	Content specified	Remarks
port	Port number of the Mirroring Controller arbitration process	The port number must be 0 to 65535. Ensure that the port number does not conflict with

Parameter	Content specified	Remarks
		other software. Do not specify an ephemeral port that may temporarily be assigned by another program.
my_address	'*ipAddrOrHostNameThatAcceptsConnectionFromMirroringControllerProcessOnDbServer*' [Setting example] my_address = '192.0.3.120'	IPv4 and IPv6 addresses can be specified. Specify the IP address, enclosed in single quotation marks (').
syslog_ident	'*programName*'	Specify using single quotation marks (') to enclose the program name used to identify the Mirroring Controller arbitration process message in the system log. Use ASCII characters excluding spaces to specify this parameter. The default is 'MirroringControllerArbiter'.
fencing_command	'*fencingCmdFilePath*' [Setting example] fencing_command = '/arbiter/fencing_dir/execute_fencing.sh'	Specify the full path of the fencing command that fences a database server where it is determined that an error has occurred. Enclose the path in single quotation marks ('). Specify the path using less than 1024 bytes.
fencing_command_timeout	Timeout for fencing command (seconds)	If the command does not respond within the specified number of seconds, it is determined that fencing has failed and a signal (SIGTERM) is sent to the fencing command execution process. Specify a value between 1 and 2147483647. The default is 20 seconds.

Information

Refer to "A.4.2 Arbitration Configuration File" for information on the parameters and for other parameters.

Windows

1. On the arbitration server, log in as any operating system user who starts and stops the arbitration process.
2. Configure the environment variables.

   Set the following environment variable:

   - PATH

     Add the installation folders "\bin" and "\lib".

   Example

   The following example configures environment variables when the installation folder is "c:\Program Files\Fujitsu \fsepv<*x*>assistant64".

   Note that "<*x*>" indicates the product version.

```
> SET PATH=c:\Program Files\Fujitsu\fsepv<x>assistant64\bin;c:\Program Files\Fujitsu
\fsepv<x>assistant64\lib;%PATH%
```

3. Create the Mirroring Controller arbitration process management directory that will store the files required by the arbitration server.

   Use ASCII characters in the Mirroring Controller arbitration process management directory.

4. In the network configuration file (network.conf), define the Mirroring Controller network configuration that will be managed by the Mirroring Controller arbitration process.

   Create network.conf in the Mirroring Controller arbitration process management directory, based on the sample file.

   Sample file

   ```
   installDir\share\mcarb_network.conf.sample
   ```

   In network.conf, specify the IP address or host name and port number of the primary server and standby server, and define the Mirroring Controller network configuration that will be managed by the Mirroring Controller arbitration process.

   Refer to "A.3 Network Configuration File" for details.

   A definition example is shown below.

   ```
   Example)
   The IDs of the servers are set to "server1" and "server2", and their port numbers are set to
   "27541".
   ```

   ```
   server1 192.0.3.100 27541
   server2 192.0.3.110 27541
   ```

5. Change the access permissions for network.conf.

   For network.conf, set read and write permissions only for the operating system user who starts and stops the arbitration process in step 1. If users other than this are granted access permissions, the mc_arb command will not work. Accordingly, users other than the operating system user who starts and stops the arbitration process in step 1 are prevented from operating the Mirroring Controller arbitration process.

   ```
   Example)
   The following is an execution example, in which the operating system user who starts and stops the
   arbitration process in step 1 is granted full access permissions as the owner when the user is
   "fsepuser". The following procedure applies when the user is logged in to the Windows server as
   "fsepuser".
   ```

   ```
   > takeown /f network.conf
   > icacls network.conf /reset
   > icacls network.conf /inheritance:r
   > icacls network.conf /grant fsepuser:F
   ```

6. In the arbitration configuration file (arbitration.conf), define the information related to control of the Mirroring Controller arbitration process.

   Create arbitration.conf in the Mirroring Controller arbitration process management directory, based on the sample file.

   Sample file

   ```
   installDir\share\mcarb_arbitration.conf.sample
   ```

   Set the parameters shown in the table below in arbitration.conf.

   Table 2.2 Parameters

Parameter	Content specified	Remarks
port	Port number of the Mirroring Controller arbitration process	The port number must be 0 to 65535. Ensure that the port number does not conflict with other software. Do not specify an ephemeral port that may temporarily be assigned by another program.

Parameter	Content specified	Remarks
my_address	'*ipAddrOrHostNameThatAcceptsConnectionFromMirroringControllerProcessOnDbServer*' [Setting example] my_address = '192.0.3.120'	IPv4 and IPv6 addresses can be specified. Enclose the parameter value in single quotation marks (').
service_name	'*registeredServiceNameOfArbitrationProcessInWindowsServices*'	Use ASCII characters excluding forward slash (/) and backslash (\) to specify this parameter. Enclose the parameter value in single quotation marks ('). The maximum length of the service name is 124 bytes.
event_source	'*eventSourceNameUsedToIdentifyArbitrationProcessMsgInEventLog*'	Use ASCII characters excluding spaces to specify this parameter. Enclose the parameter value in single quotation marks ('). The maximum length of the event source name is 255 bytes.
fencing_command	'*fencingCmdFilePath*' [Setting example] fencing_command = 'c:\\arbiter\\fencing_dir\\execute_fencing.bat'	Specify the full path of the fencing command that fences a database server where it is determined that an error has occurred. Specify "\\" as the delimiter. Enclose the path in single quotation marks ('). Specify the path using less than 260 bytes. Any multibyte characters must use the same encoding as the operating system.
fencing_command_timeout	Timeout for fencing command (seconds)	If the command does not respond within the specified number of seconds, it is determined that fencing has failed and a signal (SIGTERM) is sent to the fencing command execution process. Specify a value between 1 and 2147483647. The default is 20 seconds.

Information

Refer to "A.4.2 Arbitration Configuration File" for information on the parameters and for other parameters.

7. Change the access permissions for arbitration.conf.

   For arbitration.conf, set read and write permissions only for the operating system user who starts and stops the arbitration process in step 1. If users other than this are granted access permissions, the mc_arb command will not work.

   ```
   Example)
   The following is an execution example, in which the operating system user who starts and stops the
   arbitration process in step 1 is granted full access permissions as the owner when the user is
   "fsepuser". The following procedure applies when the user is logged in to the Windows server as
   "fsepuser".
   ```

```
> takeown /f arbitration.conf
> icacls arbitration.conf /reset
> icacls arbitration.conf /inheritance:r
> icacls arbitration.conf /grant fsepuser:F
```

8. Configure Windows Firewall.

   If Windows Firewall is used, enable the port number of Mirroring Controller specified in the network configuration file in step 3. Refer to "E.2 Windows Firewall Settings" for details.

9. Register the Mirroring Controller arbitration process as a Windows service.

   Execute the mc_arb command in register mode.

   For the -P option of the mc_arb command, specify the password of the operating system user who executes the command.

   Example)

```
> mc_arb register -M D:\mcarb_dir\arbiter1 -P ********
```

Note

When specifying the password in the -P option of the mc_arb command, for security reasons, you should be careful not to allow other users to access it.

Information

Use the mc_arb command with the -S option to specify automatic start and stop of the Mirroring Controller arbitration process. Refer to "2.13 Setting Automatic Start and Stop of the Mirroring Controller Arbitration Process" for details.

The Mirroring Controller arbitration process is registered as a Windows service using the service name specified in the service_name parameter of arbitration.conf in step 6.

You can execute the sc qc command to check the registration status.

2.3.2 Creating a User Exit for the Arbitration Server

The only user exit for the arbitration server is the fencing command.

The fencing command is a user command that is called by the Mirroring Controller arbitration process if Mirroring Controller performs arbitration processing and determines that a database server is unstable.

In the fencing command, the user implements a process that isolates a database server from a cluster system by, for example, stopping the target operating system or server. The fencing command that was created is to be specified for the parameter in the arbitration configuration file. Refer to "A.4.2 Arbitration Configuration File" for information on the parameters.

- Fencing the primary server during the switch
  - Prevent the Mirroring Controller management process on the primary server from communicating with the Mirroring Controller management process on the other server.
  - Prevent applications from connecting to the primary server instance.
- Fencing the standby server during disconnection
  - Prevent the Mirroring Controller management process on the standby server from communicating with the Mirroring Controller management process on the other server.
  - Prevent applications from connecting to the standby server instance.
  - Prevent the standby server from continuing streaming replication.

See

Refer to "Appendix C User Commands" for information on user exits.

2.3.3 Starting the Mirroring Controller Arbitration Process

This section explains how to start the Mirroring Controller arbitration process.

An operating system user who has logged in to the arbitration server can start the Mirroring Controller arbitration process by executing the mc_arb command in start mode.

Linux

Example)

```
$ mc_arb start -M /mcarb_dir/arbiter1
```

Windows

Example)

```
> mc_arb start -M D:\mcarb_dir\arbiter1
```

2.4 Setting Up the Primary Server

This section explains how to set up the primary server.

2.4.1 Setting Up Database Multiplexing Mode on the Primary Server

This section explains how to set up database multiplexing mode on the primary server.

In database multiplexing, the files that are required for operations are managed in the Mirroring Controller management directory.

There is one Mirroring Controller management directory for each instance.

Note

- Do not place the Mirroring Controller management directory in a directory managed by Fujitsu Enterprise Postgres, otherwise it may be deleted by mistake with the directories managed by Fujitsu Enterprise Postgres, and an old version of files may be restored.

See

- Refer to "Preparing Directories for Resource Deployment" in the Installation and Setup Guide for Server for details on the directories that are managed by Fujitsu Enterprise Postgres.
- Refer to "mc_ctl" in Reference for information on the command.
- Refer to "Appendix A Parameters" for details on each parameter to be edited for the setup.

Perform the following procedure:

1. Log in to the primary server.
2. Create the Mirroring Controller management directory that will store the files required by database multiplexing.

   Use ASCII characters in the Mirroring Controller management directory.

   Additionally, grant "Write" permission to the instance administrator user for the Mirroring Controller management directory.

3. In the network configuration file (network.conf), define the network configuration that will link between the Mirroring Controller processes.

   Create the network.conf file in the Mirroring Controller management directory, based on the sample file. For network.conf, set read and write permissions for the instance administrator user only.

   If users other than the instance administrator user are granted access, the mc_ctl command will not work. In this way, users other than the instance administrator user are prevented from operating Mirroring Controller.

   Sample file

   ```
   /installDir/share/mc_network.conf.sample
   ```

   In network.conf, specify the IP address or host name and port number of the primary server and standby server, and define the network configuration that will link between the Mirroring Controller processes, and between Mirroring Controller processes and the Mirroring Controller arbitration process.

   Refer to "A.3 Network Configuration File" for details.

   A definition example is shown below.

   The content to be defined depends on the operation settings at the time a heartbeat abnormality is detected.

   When automatic degradation by the arbitration server is selected

   ```
   Example)
   The IDs of the primary server and standby server are set to "server1" and "server2", and their
   port numbers are set to "27540" and "27541". The ID of the server of the Mirroring Controller
   arbitration process is set to "arbiter", and its port number is set to "27541".
   ```

   ```
   server1 192.0.2.100,192.0.3.100 27540,27541 server
   server2 192.0.2.110,192.0.3.110 27540,27541 server
   arbiter 192.0.3.120 27541 arbiter
   ```

   Ensure that the port numbers set for the primary server, standby server, and arbitration server do not conflict with other software. In addition, when the arbitration server is used for automatic degradation, use a network in which the arbitration network is not affected by a line failure in the admin network.

   When the server type is "server", two IP addresses or host names, and two port numbers need to be specified in the following order:

   - IP address or host name of the database server used as the admin network
   - IP address or host name of the database server used as the arbitration network
   - Port number of the database server used as the admin network
   - Port number of the database server used as the arbitration network

   If the server type is "arbiter", specify the IP address or host name set for the my_address parameter and the port number set for the port parameter in arbitration.conf of the arbitration server.

   When operation other than automatic degradation by the arbitration server is selected

   ```
   Example)
   The IDs of the servers are set to "server1" and "server2", and their port numbers are set to
   "27540".
   ```

   ```
   server1 192.0.2.100 27540
   server2 192.0.2.110 27540
   ```

   Ensure that the port numbers for the primary and standby server do not conflict with other software.

   Register in /etc/services the port number of the primary server, because programs such as WebAdmin use it to search for available port numbers.

   Register any name as the service name.

4. Define the information related to Mirroring Controller monitoring and control in the *serverIdentifier*.conf file.

   Create the *serverIdentifier*.conf file in the Mirroring Controller management directory, based on the sample file.

For *serverIdentifier*.conf, set read and write permissions for the instance administrator user only. If users other than the instance administrator user are granted access, the mc_ctl command will not work.

As the file name for the *serverIdentifier*.conf file, use the server identifier name that was specified in the network.conf file in step 3.

Sample file

```
/InstallDir/share/mc_server.conf.sample
```

Set the parameters shown in the table below in the *serverIdentifier*.conf file.

Table 2.3 Parameters

Parameter	Content specified	Remarks
db_instance	'*dataStorageDestinationDir*'	Use ASCII characters, enclosed in single quotation marks (').
db_instance_password	'*passwordOfInstanceAdminUser*'	If password authentication is performed, you must specify this parameter in the settings used when Mirroring Controller connects to a database instance. Use ASCII characters, enclosed in single quotation marks ('). If the specified value of this parameter includes ' or \, write \' or \\, respectively.
enable_hash_in_password	on or off	Specify on to treat the # in the db_instance_password specification as a password character, or off to treat it as a comment. The default is "off".
syslog_ident	'*programName*'	Specify the program name to be used to identify the Mirroring Controller messages in the system log. Use ASCII characters excluding spaces, enclosed in single quotation marks ('). Use the same program name as the parameter in the postgresql.conf file ensures that the Mirroring Controller output content can be referenced transparently, so log reference is easy.
remote_call_timeout	Admin communication timeout	Specify the timeout value (milliseconds) of the Mirroring Controller agent process for communication between servers. Specify a value that is less than the operation system TCP connection timeout. Also, when using the Mirroring Controller arbitrage process for arbitrage, fencing, and state transition commands, specify a value that is greater than the sum of the timeout values.
heartbeat_error_action	Operation when a heartbeat abnormality is detected using operating system or server heartbeat monitoring	arbitration: Perform automatic degradation using the arbitration server. command: Call a user exit to determine degradation, and perform automatic degradation if required. message: Notify messages.

Parameter	Content specified	Remarks
		fallback: Perform automatic degradation unconditionally. Set the same value on the primary server and standby server.
heartbeat_interval	Interval time for abnormality monitoring during heartbeat monitoring of the operating system or server (milliseconds)	Abnormality monitoring of the operating system or server is performed at the interval (milliseconds) specified in heartbeat_interval. This parameter setting is used as the default for database process heartbeat monitoring, streaming replication abnormality monitoring, and disk abnormality monitoring. When setting the monitoring time, there are some considerations to take into account to optimize degradation using abnormality monitoring. Refer to "2.11.4.1 Tuning for Abnormality Monitoring of the Operating System or Server" for details.
heartbeat_timeout	Timeout for abnormality monitoring during heartbeat monitoring of the operating system or server (seconds)	
heartbeat_retry	Number of retries for abnormality monitoring during heartbeat monitoring of the operating system or server (number of times)	
fencing_command	'*fencingCmdFilePath*' [Setting example] fencing_command = '/mc/fencing_dir/execute_fencing.sh'	Specify the full path of the fencing command that fences a database server where an error is determined to have occurred. Enclose the path in single quotation marks ('). This parameter must be specified when "command" is set for heartbeat_error_action. Specify the path using less than 1024 bytes.
fencing_command_timeout	Fencing command timeout (seconds)	If the command does not respond within the specified number of seconds, fencing is determined to have failed and a signal (SIGTERM) is sent to the fencing command execution process. Specify a value between 1 and 2147483647. The default is 20 seconds.
arbitration_timeout	Timeout for arbitration processing in the Mirroring Controller arbitration process (seconds)	The specified value must be at least equal to the heartbeat monitoring time of the operating system or server + fencing_command_timeout in the arbitration configuration file. If there is no response for at least the number of seconds specified, the primary server will not be switched and the standby server will not be disconnected. Therefore, perform degradation manually. Specify a value between 1 and 2147483647. This parameter does not need to be set for operation that does not use the arbitration server.
arbitration_command	'*arbitrationCmdFilePath*' [Setting example]	Specify the full path of the arbitration command to be executed when an abnormality is detected during heartbeat monitoring of the operating system or server.

Parameter	Content specified	Remarks
	arbitration_command = '/mc/arbitration_dir/execute_arbitration_command.sh'	Enclose the path in single quotation marks ('). This parameter must be specified when "command" is set for heartbeat_error_action. Specify the path using less than 1024 bytes.
arbitration_command_timeout	Timeout for arbitration command (seconds)	If the arbitration command does not respond within the specified number of seconds, it is determined that execution of the arbitration command has failed and a signal (SIGTERM) is sent to the arbitration command execution process. Specify a value between 1 and 2147483647. This parameter can be specified only when "command" is set for heartbeat_error_action.

Information

Refer to "A.4.1 Server Configuration File for the Database Servers" for information on the parameters and for other parameters.

2.4.2 Creating, Setting, and Registering the Primary Server Instance

This section explains how to create, set, and register the primary server instance.

See

- Refer to "Client Authentication" in the PostgreSQL Documentation for information on the pg_hba.conf file.
- Refer to "A.1 Parameters Set on the Primary Server" for information on the postgresql.conf file.
- Refer to "mc_ctl" in Reference for information on the command.

Perform the following procedure:

1. Refer to "Setup" in the Installation and Setup Guide for Server, and then perform the Fujitsu Enterprise Postgres setup and create the Fujitsu Enterprise Postgres instance.

   Use ASCII characters in the data storage destination directory.

   Note

   - If degradation starts occurring due to an error during operations in database multiplexing mode, recovery is required for the standby server. There are some conditions to execute the pg_rewind command to recover the standby server. One of the conditions can be satisfied by enabling checksums when executing the initdb command. This is not mandatory. Refer to "4.1.1.1.3 Identify cause of error and perform recovery" for details.

2. When using transparent data encryption, configure the encryption settings for the storage data.

   If you want to use a file-based keystore, create a keystore file.

   If you want to use the key management system as a keystore, set the connection information for the key management system and declare the master encryption key.

   Refer to "Protecting Storage Data Using Transparent Data Encryption" or "Using Transparent Data Encryption with Key Management Systems as Keystores" in the Operation Guide for details, and then configure the settings.

3. Add the following entry to the pg_hba.conf file to authenticate connections from the standby server.

   Copy the file to the standby server later.

```
# TYPE    DATABASE        USER        ADDRESS                     METHOD
  host    replication     fsep        standbyServerAddress        authenticationMethod
  host    replication     fsep        primaryServerAddress        authenticationMethod
```

   For the primary and standby server addresses, specify the IP address that will connect to the log transfer network.

   Additionally, all servers can be used as the primary server or the standby server, so add entries for the addresses of all servers that comprise the database multiplexing system.

Point

Setting an authentication method other than trust authentication

If the primary server becomes the standby server, to perform automatic authentication of connections to the primary server, create the .pgpass file in the home directory of the instance administrator user, and then specify a password for the replication database. Accordingly, the instance administrator operating system user and the user registered in the database will be the same, so you can verify that the connection was not made by an unspecified user. Additionally, the password that was set beforehand will be used in the authentication, so that the connection will be automatic.

Note

If trust authentication is set, all OS users who can log in to the primary server will be able to connect, and if one of these is a malicious user, then that user can corrupt the standby server data, or cause the job system to fail, by sending an erroneous transaction log. Therefore, decide on the authentication method according to the security requirements of the system using database multiplexing mode.

Refer to "Authentication Methods" in the PostgreSQL Documentation for details on the authentication methods that can be set.

4. Configure this setting to enable the instance administrator user of the primary server to connect as a database application.

   This setting enables the connection to the instance using the user name of the instance administrator user, so that Mirroring Controller can monitor instance errors. Configure this setting to enable the connection to the postgres database.

   - If password authentication is used

     In the db_instance_password parameter of the *serverIdentifier*.conf file, specify the password for the instance administrator user. This password is used to connect to the database instance. If a password is not specified in the db_instance_password parameter, the connection to the database instance from Mirroring Controller will fail, and it will not be possible to perform the process monitoring of the instance.

   - If password authentication is not used

     There is no need to specify the password in the db_instance_password parameter.

     Even if the password for the instance administrator user is specified in the db_instance_password parameter, it will be ignored.

   - If certificate authentication using SSL is used

     Specify connection parameters for SSL in the db_instance_ext_ pq_conninfo parameter and db_instance_ext_jdbc_conninfo parameter in the *serverIdentifier*.conf file. If the pasrameters are not specified, the connection to the database instance from Mirroring Controller will fail, and it will not be possible to perform the process monitoring of the instance. If certificate authentication using SSL is not performed, the parameters specification is not required.

   An example of setting the authentication method is shown below.

```
# TYPE    DATABASE        USER        ADDRESS         METHOD
  host    postgres        fsep        127.0.0.1/32    authenticationMethod
```

Note

Mirroring Controller uses the PostgreSQL JDBC 4.2 driver to connect to the database instance. Therefore, for the authentication method, specify a method supported by the JDBC driver. If an authentication method not supported by the JDBC driver is specified, Mirroring Controller will fail to start. Refer to the PostgreSQL JDBC Driver Documentation for information on authentication methods supported by the JDBC driver.

5. To use database multiplexing mode, specify the parameters shown in the table below in the postgresql.conf file.

   The postgresql.conf file is copied when the standby server instance is created. Accordingly, set the required parameters in the standby server.

   To use database multiplexing mode, specify the parameters shown in the table below in the postgresql.conf file. After editing the postgresql.conf file, restart the instance.

Table 2.4 Parameters

Parameter	Content specified	Remarks
wal_level	replica or logical	Specify "logical" when logical decoding is also to be used.
max_wal_senders	2 or more	Specify "2" when building a Mirroring Controller cluster system. When additionally connecting asynchronous standby servers to the cluster system, add the number of simultaneous connections from these standby servers.
synchronous_standby_names	'*standbyServerName*'	Specify the name that will identify the standby server. Enclose the name in single quotation marks ('). Do not change this parameter while Mirroring Controller is running. Do not specify multiple names to this parameter as the Mirroring Controller can manage only one standby server.
hot_standby	on	
wal_keep_size	WAL save size (megabytes)	If a delay exceeding the value set in this parameter occurs, the WAL segment required later by the primary server may be deleted. Additionally, if you stop a standby server (for maintenance, for example), consider the stop time and set a value that will not cause the WAL segment to be deleted. Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on estimating the WAL save size.
wal_log_hints	on	When using the pg_rewind command to recover a standby server, specify this parameter or enable checksums when executing the initdb command.
wal_sender_timeout	Timeout (milliseconds)	Specify the time period after which it is determined that an error has occurred in the transaction log transfer on the primary server. By aligning this value with the value for the database process heartbeat monitoring time, you can unify the time after which it is determined that an error has occurred.
archive_mode	on	Specify the archive log mode.
archive_command	'*installDir*/bin/pgx_walcopy.cmd "%p"	Specify the command and storage destination to save the transaction log.

Parameter	Content specified	Remarks
	"*backupDataStorageDestinationDirectory*/archived_wal/%f"'	
backup_destination	Backup data storage destination directory	Specify the name of directory where to store the backup data. Set the permissions so that only the instance administrator user can access the specified directory. Specify the same full path on all servers, so that the backup data of other servers can be used to perform recovery.
max_connections	Number of simultaneous client connections to the instance + superuser_reserved_connections	The value specified is also used to restrict the number of connections from client applications and the number of connections for the management of instances. Refer to "When an Instance was Created with the initdb Command" in the Installation and Setup Guide for Server, and "Connections and Authentication" in the PostgreSQL Documentation, for details.
superuser_reserved_connections	Add the number of simultaneous executions of mc_ctl status (*1) + 2	Specify the number of connections reserved for connections from database superusers. Add the number of connections from Mirroring Controller processes. Also reflect the added value in the max_connections parameter.
wal_receiver_timeout	Timeout (milliseconds)	Specify the time period after which it is determined that an error has occurred when the transaction log was received on the standby server. By aligning this value with the value for the heartbeat monitoring time of the database process, you can unify the time after which it is determined that an error has occurred.
restart_after_crash	off	If "on" is specified, or the default value is used for this parameter, behavior equivalent to restarting Fujitsu Enterprise Postgres, including crash recovery, will be performed when some server processes end abnormally. However, when database multiplexing monitoring is used, a failover will occur after an error is detected when some server processes end abnormally, and the restart of those server processes is forcibly stopped. Specify "off" to prevent behavior such as this from occurring for no apparent reason.
synchronous_commit	on or remote_apply	Specify up to what position WAL send is to be performed before transaction commit processing returns a normal termination response to a client. The recommended value is "on" or "remote_apply" to prevent data loss caused by operating system or server down immediately after a switch or switch.
recovery_target_timeline	latest	Specify "latest" so that the new standby server (original primary server) will follow the new primary server when a switch occurs. This parameter is required when the original primary server is incorporated as a new standby server after the primary server is switched.

*1: Number of simultaneous executions of the mc_ctl command in the status mode.

2.4.3 Starting Mirroring Controller on the Primary Server

This section explains how to start Mirroring Controller on the primary server.

When the arbitration server is used for automatic degradation, start the Mirroring Controller arbitration process on the arbitration server in advance.

1. Start the Mirroring Controller process.

   Enabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode.

   Example)

   ```
   $ mc_ctl start -M /mcdir/inst1
   ```

   Disabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

   Example)

   ```
   $ mc_ctl start -M /mcdir/inst1 -F
   ```

   Note

   - When the arbitration server is used for automatic degradation, the database server must connect to the arbitration server, and as a result, Mirroring Controller startup may take longer than when the arbitration server is not used.
   - If the parameter for heartbeat monitoring of operating systems or servers set by the abitration server is greater than parameter for heartbeat monitoring of operating systems and servers of the Mirroring Controller, the Mirroring Controller may fail to start. In this case, check the contents of the message notification and review the parameters for heartbeat monitoring of operating systems or servers for the arbitration server or Mirroring Controller.
   - If the heartbeat_error_action parameter in *serverIdentifier*.conf is set to "message", even if automatic switch/disconnection is enabled and Mirroring Controller is started, only message output is performed when a heartbeat abnormality is detected during heartbeat monitoring of operating systems and servers - switch/disconnection is not performed.
   - Mirroring Controller startup usually fails if the standby server is mistakenly started as the primary server or if the old primary server is not recovered after the switch and is then mistakenly started as the primary server. However, if the admin network is disconnected, then startup does not fail, and both servers may become primary servers. Therefore ensure that the admin network is connected before starting Mirroring Controller.

   Point

   - The mc_ctl command fails if the Mirroring Controller arbitration process has not been started on the arbitration server when the arbitration server is used for automatic degradation. However, if the Mirroring Controller arbitration process cannot be started in advance, it can be started by specifying the --async-connect-arbiter option in the mc_ctl command.
   - After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

2. Obtain the backup.

   Use the pgx_dmpall command to collect the backup.

2.5 Setting Up the Standby Server

This section explains how to set up the standby server.

2.5.1 Setting Up Database Multiplexing Mode on the Standby Server

This section explains how to set up database multiplexing mode on the standby server.

In database multiplexing, the files that are required for operations are managed in the Mirroring Controller management directory.

There is one Mirroring Controller management directory for each instance.

Note

- Do not place the Mirroring Controller management directory in a directory managed by Fujitsu Enterprise Postgres, otherwise it may be deleted by mistake with the directories managed by Fujitsu Enterprise Postgres, and an old version of files may be restored.
- When creating a standby server for a large database, stop job system operations, specify a large value for the wal_keep_size parameter, or use replication slots.
  This is because WALs generated after the standby server is built using the pg_basebackup command, but before it is started, need to be retained. However, the number of WAL segments that can be retained is constrained by the wal_keep_size parameter.
  Additionally, setting the wal_keep_size parameter requires consideration regarding stabilization of the database multiplexing mode (refer to "2.11.1 Tuning to Stabilize the Database Multiplexing Mode" for details).

See

- Refer to "Preparing Directories for Resource Deployment" in the Installation and Setup Guide for Server for details on the directories that are managed by Fujitsu Enterprise Postgres.
- Refer to "pg_basebackup" in "Reference" in the PostgreSQL Documentation for information on the pg_basebackup command.
- Refer to "mc_ctl" in Reference for information on the command.
- Refer to "Appendix A Parameters" for details on each parameter to be edited for the setup.
- Refer to "Replication Slots" in the PostgreSQL Documentation for information on replication slots.

Perform the following procedure:

1. Log in to the standby server.
2. Create the Mirroring Controller management directory that will store the files required by database multiplexing.

   Use ASCII characters in the Mirroring Controller management directory.

   Additionally, grant "Write" permission to the instance administrator user for the Mirroring Controller management directory.
3. Copy, and then deploy, the network.conf file of the primary server.

   Copy the network.conf file that was defined in the primary server setup, and deploy it to the Mirroring Controller management directory of the standby server.

   Set read and write permissions for the instance administrator user only. If users other than the instance administrator user are granted access, the mc_ctl command will not work. Accordingly, users other than the instance administrator user are prevented from operating Mirroring Controller.

   Register in /etc/services the port number of the standby server that was specified in the network.conf file, because programs such as WebAdmin use it to search for available port numbers.

   Register any name as the service name.

4. Copy, and then deploy, the *serverIdentifier*.conf file of the primary server.

   Copy the *serverIdentifier*.conf file that was defined in the primary server setup, and deploy it to the Mirroring Controller management directory of the standby server.

   Set read and write permissions for the instance administrator user only. If users other than the instance administrator user are granted access permissions, the mc_ctl command will not work.

2.5.2 Creating, Setting, and Registering the Standby Server Instance

This section explains how to create, set, and register the standby server instance.

See

- Refer to "Appendix A Parameters" for details on each parameter.
- Refer to "mc_ctl" in Reference for information on the command.

Perform the following procedure:

1. Set the kernel parameters.

   Refer to "Configuring Kernel Parameters" in the Installation and Setup Guide for Server for details.

2. When using transparent data encryption, configure the encryption settings for the storage data.

   Refer to "Protecting Storage Data Using Transparent Data Encryption" or "Using Transparent Data Encryption with Key Management Systems as Keystores" in the Operation Guide for details, and then configure the settings.

3. Execute the pg_basebackup command to create a copy of the primary server instance on the standby server.

   Example)

   ```
   $ pg_basebackup -D /database/inst1 -X fetch --waldir=/transaction/inst1 --progress --verbose -R
   --dbname='application_name=standbyServerName' -h primaryServerIpAddress -p
   primaryServerPortNumber
   ```

   ### Note

   - Use the pg_basebackup command with the -R option to create a standby.signal file. If you do not create the standby.signal file, the Mirroring Controller cannot be started as a standby server.
   - If using a method that requires password authentication for connections to the primary server, you will need to ensure that authentication is performed automatically. If the -R option is specified for the pg_basebackup command and the password parameter is specified for the --dbname option, the pg_basebackup command will set the password in the primary_conninfo parameter in postgresql.auto.conf file, enabling connections to be performed automatically.

     If a password is not set in the primary_conninfo parameter in postgresql.auto.conf file, it will be necessary to create a .pgpass file in the home directory of the instance administrator user, and specify a password for the replication database.
   - The primary_conninfo parameter should not be set in the postgresql.conf file, but only in the postgresql.auto.conf file using the pg_basebackup command.
   - When executing the pg_basebackup command, consider the following for collection of transaction logs.
     - When "fetch" is specified for the -X option of the command

       Transaction logs are collected at the end of the backup, so it is necessary to ensure that transaction logs that occur during backup are not deleted from the primary server. Therefore, allow for a sufficient value for the wal_keep_size parameter in postgresql.conf.

     - When the -X option is omitted or "stream" is specified for the -X option of the command

       Transaction logs are streamed, so when Mirroring Controller is running on the primary server, the connection is changed to a synchronous standby server on detection of a streaming replication connection using this command. Therefore, if a job has

started on the primary server, the primary server will be impacted, therefore execute this command after stopping only the Mirroring Controller process on the primary server.

See

Refer to "Hot Standby" in the PostgreSQL Documentation for information on the standby.signal file.

4. Set the parameters shown in the table below in the postgresql.conf file.

Table 2.5 Parameters

Parameter	Content specified	Remarks
synchronous_standby_names	'*primaryServerName*'	Required after switching the primary server and then changing the original primary server to the new standby server. Enclose the name in single quotation marks ('). Do not change this parameter while Mirroring Controller is running. Do not specify multiple names to this parameter as the Mirroring Controller can manage only one standby server.

2.5.3 Starting Mirroring Controller on the Standby Server

This section explains how to start Mirroring Controller on the standby server.

When the arbitration server is used for automatic degradation, start the Mirroring Controller arbitration process on the arbitration server in advance.

1. After ensuring that the Mirroring Controller process of the primary server has started, start Mirroring Controller on the standby server.

   Enabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode with the -f option specified. This action enables automatic switch/disconnection.

   If you start Mirroring Controller and the instance without specifying the -f option, automatic switch/disconnection will not be enabled. To enable both, start Mirroring Controller and then execute the mc_ctl command in enable-failover mode or restart Mirroring Controller with the -f option specified.

   Example)

```
$ mc_ctl start -M /mcdir/inst1
```

   Disabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

   Example)

```
$ mc_ctl start -M /mcdir/inst1 -F
```

2. Check the status of the Mirroring Controller process.

   As the instance administrator user, execute the mc_ctl command in status mode. Ensure that "mirroring status" is switchable.

   Example)

```
$ mc_ctl status -M /mcdir/inst1
```

Note

- When the arbitration server is used for automatic degradation, the time required for the database server to connect to the arbitration server is added on. Therefore, Mirroring Controller startup may take longer than when the arbitration server is not used.
- If the parameter for heartbeat monitoring of operating systems or servers set by the abitration server is greater than parameter for heartbeat monitoring of operating systems and servers of the Mirroring Controller, the Mirroring Controller may fail to start. In this case, check the contents of the message notification and review the parameters for heartbeat monitoring of operating systems or servers for the arbitration server or Mirroring Controller.
- If the heartbeat_error_action parameter in *serverIdentifier*.conf is set to "message", even if automatic switch/disconnection is enabled and Mirroring Controller is started, only message output is performed when a heartbeat abnormality is detected during heartbeat monitoring of operating systems and servers - switch/disconnection is not performed.
- Mirroring Controller startup usually fails if the standby server is mistakenly started as the primary server or if the old primary server is not recovered after the switch and is then mistakenly started as the primary server. However, if the admin network is disconnected, then startup does not fail, and both servers may become primary servers. Therefore, ensure that the admin network is connected before starting Mirroring Controller.

Point

- The mc_ctl command fails if the Mirroring Controller arbitration process has not been started on the arbitration server when the arbitration server is used for automatic degradation. However, if the Mirroring Controller arbitration process cannot be started in advance, it can be started by specifying the --async-connect-arbiter option in the mc_ctl command.
- After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

2.6 Creating a User Exit for a Database Server

This section explains how to create a user exit for a database server.

The user command types explained below can be used as user exits. These commands are called by Mirroring Controller management processes.

The user can create user exits as required.

Specify the user commands that were created for the parameters in the server configuration file of the database server. Refer to "A.4.1 Server Configuration File for the Database Servers" for information on these parameters.

User command types

- Fencing command

  This user command performs fencing if Mirroring Controller performs arbitration processing and determines that a database server is unstable.

- Arbitration command

  This user command performs arbitration processing in lieu of the arbitration server when there is no arbitration server.

- State transition commands

  These user commands are called when Mirroring Controller performs state transition of a database server.
  It includes the following types:

  - Post-switch command

    This user command is called after a promotion from standby server to primary server.

- Pre-detach command

  This user command is called before the standby server is disconnected from a cluster system.
  If the pre-detach command is specified on both the primary server and standby server, it is called first on the standby server and then on the primary server.

  If the settings are configured to forcibly stop the instance on the standby server when the standby server is disconnected, the pre-detach command is called on the standby server and then the instance on the standby server is stopped.

- Post-attach command

  This user command is called after the standby server has been attached to a cluster system.
  If the post-attach command is specified on both the primary server and standby server, it is called first on the primary server and then on the standby server.

Point

When the arbitration server is used for automatic degradation and the requirements can be satisfied using the fencing command on the arbitration server only, the fencing command on the database server is not required. In addition, if the requirements can be satisfied using the fencing command on the database server only, create a fencing command on the arbitration server for termination processing only (without implementation).

Table 2.6 Availability of user commands, and database server calling the command

User command	Operation when a heartbeat abnormality is detected using operating system or server heartbeat monitoring				Database server calling the command	
	Message output	Automatic degradation by arbitration server	Automatic degradation by arbitration command	Unconditional automatic degradation	Primary server	Standby server
Fencing command	Y (*1)	Y (*2)	R	N	Y	Y
Arbitration command	N	N	R	N	Y	Y
Post-switch command	Y	Y	Y	Y	Y	N
Pre-detach command	Y	Y	Y	Y	Y	Y (*3)
Post-attach command	Y	Y	Y	Y	Y	Y (*3)

R: Required

Y: Can be used

N: Cannot be used

*1: Called only when the mc_ctl command is used to execute forced switching or forced disconnection.

*2: Creation of a fencing command on a database server is optional, but it must be created on the arbitration server.

*3: If message output or unconditional automatic degradation is selected, this command is called only from the primary server.

See

Refer to "Appendix C User Commands" for information on the interface for each user command.

2.7 Confirming the Streaming Replication Status

Before performing the setup of the database multiplexing mode, ensure that the prerequisite streaming replication feature has been set up correctly.

Perform the following procedure:

1. On the primary server, ensure that single-row searches can be performed using the pg_stat_replication statistics view.

   An example output of the psql command is shown below.

   Example)

```
postgres=# select * from pg_stat_replication;
-[ RECORD 1 ]----+------------------------------
pid              | 10651
usesysid         | 10
usename          | fsep
application_name | standby
client_addr      | 192.0.2.210
client_hostname  |
client_port      | 55098
backend_start    | 2022-03-23 11:17:49.628793+09
backend_xmin     |
state            | streaming
sent_lsn         | 0/3000060
write_lsn        | 0/3000060
flush_lsn        | 0/3000060
replay_lsn       | 0/3000060
write_lag        |
flush_lag        |
replay_lag       |
sync_priority    | 1
sync_state       | sync
reply_time       | 2022-03-23 11:23:27.703366+09
```

2. Confirm the search results of step 1.

   Ensure that the connection established with the intended standby server is in synchronous mode.

   Table 2.7 Items to be checked

Item	Required value
application_name	Value specified for synchronous_standby_names parameter in the postgresql.conf file of the primary server.
client_addr	IP address of the standby server.
state	"streaming".
sync_state	"sync".

See

- Refer to "The Statistics Collector" in "Server Administration" in the PostgreSQL Documentation for information on the pg_stat_replication statistics view.
- Note that the pg_stat_replication statistics view may change in the future.

2.8 Checking the Connection Status

This section explains how to check the connection status from a database server or the arbitration server.

2.8.1 Checking the Connection Status on a Database Server

This section explains how to use a database server to check the connection status of the Mirroring Controller arbitration process and the Mirroring Controller process on the primary server and standby server.

Perform the following procedure:

1. On the primary server and standby server, execute the mc_ctl command in status mode with the --arbiter option specified.

   Example)
   The mc_ctl command is executed with the --arbiter option specified, and the status is output.

```
$ mc_ctl status --arbiter -M /mcdir/inst1

arbiter_id   host            status
-------------------------------------
arbiter      192.0.3.120     online
```

2. On the primary server and standby server, check the result displayed by executing the mc_ctl command in status mode in step 1.

   Items to be checked

   Check that the output status" is "online".

See

Refer to the Reference for information on the mc_ctl command.

2.8.2 Checking the Connection Status on the Arbitration Server

This section explains how to use the arbitration server to check the connection status of the Mirroring Controller arbitration process and the Mirroring Controller process on the primary server and standby server.

Perform the following procedure:

1. Execute the mc_arb command in status mode on the arbitration server.

   The example below executes the mc_arb command, and shows the status.

   Linux

   Example)

```
$ mc_arb status -M /mcarb_dir/arbiter1

server_id    host            status
--------------------------------------
server1      192.0.3.100     online
server2      192.0.3.110     online
```

   Windows

   Example)

```
> mc_arb status -M D:\mcarb_dir\arbiter1

server_id    host            status
--------------------------------------
server1      192.0.3.100     online
server2      192.0.3.110     online
```

2. Check the result displayed by executing the mc_arb command in step 1.

   Items to be checked

   Check that the output status is "online" on both lines.

See

Refer to the Reference for information on the mc_arb command.

2.9 Creating Applications

This section explains how to create applications using database multiplexing, and points that should be noted when you create the applications.

2.9.1 Application Connection Server Settings

If database multiplexing is used and a failover occurs, it will be necessary to switch the application connection server. Accordingly, use the application connection switch feature to create applications.

See

Refer to "Application Connection Switch Feature" in the Application Development Guide for details.

2.10 Checking the Behavior

To check if the environment setup was performed correctly, start the application and then check the behavior of the switch and rebuild.

2.11 Tuning

This section explains how to tune database multiplexing mode.

2.11.1 Tuning to Stabilize the Database Multiplexing Mode

When large amounts of data are updated, the write-to load for the database will become great, and the multiplexing state may become unstable.

Accordingly, by editing the parameters below in the postgresql.conf file, a stable multiplexing state can be maintained. Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on transaction log space requirements.

Table 2.8 Parameters

Parameter	Content
wal_keep_size	Refer to "2.4.2 Creating, Setting, and Registering the Primary Server Instance" for details.
max_wal_size	The transaction log is written out according to the checkpoint trigger. If a transaction log with the capacity of the value specified in this parameter is generated, the checkpoint will be executed. If a large value is specified in this parameter, the time required for crash recovery will increase. If a small value is specified in this parameter, many checkpoints will be generated, which will affect the performance of the applications that connect to the primary server.

2.11.2 Tuning to Stabilize Queries on the Standby Server

Queries made using reference jobs on the standby server may be canceled by jobs executed on the primary server.

To reduce the possibility of a job being canceled, specify as large a value as possible for the max_standby_archive_delay parameter in the postgresql.conf file.

See

- Refer to "Handling Query Conflicts" in the PostgreSQL Documentation for details.
- Refer to "Standby Servers" in the PostgreSQL Documentation for details on the max_standby_archive_delay parameter.

2.11.3 Tuning to Stabilize Queries on the Standby Server (when Performing Frequent Updates on the Primary Server)

If jobs are updated on the primary server regularly and frequently, it will be easy for the query made by the reference job on the standby server to be canceled. In this case, edit one of the postgresql.conf file parameters shown in the table below.

Table 2.9 Parameters

Parameter	Description
hot_standby_feedback	When "on" is set, the deletion (vacuum) of the data area that was deleted or updated on the primary server is suppressed. Accordingly, the query on the standby server will not be canceled. (*1)
vacuum_defer_cleanup_age	The deletion (vacuum) of the data area that was deleted or updated on the primary server is delayed until the specified number of transactions is processed. Accordingly, the probability that the query on the standby server will be canceled decreases.

*1: Because the vacuum is delayed, the data storage destination disk space of the primary server comes under pressure.
Additionally, if there is conflict between accesses and queries executed on the standby server, transaction logs indicating this conflict will be transferred.
Accordingly, specify as large a value as possible for the max_standby_archive_delay parameter so that access conflicts do not occur.

See

- Refer to "Standby Servers" in the PostgreSQL Documentation for details on the hot_standby_feedback parameter.
- Refer to "Primary Server" in the PostgreSQL Documentation for details on the vacuum_defer_cleanup_age parameter.

2.11.4 Tuning for Optimization of Degradation Using Abnormality Monitoring

Mirroring Controller uses a monitoring method that outputs an error if the timeout or number of retries is exceeded when accessing resources targeted for monitoring. Setting inappropriate values in these settings may lead to misdetection or a delay in automatic degradation, so you must design these values appropriately.

For example, the following type of issue occurs if the tuning related to abnormality monitoring is not performed appropriately.

- If the timeout is too short

  Results in redundant degradation and availability falls.

- If the timeout is too long

  It takes longer for automatic degradation to be performed even when an error affecting operational continuity occurs, potentially causing downtime.

You can optimize degrading operation by editing the values for the parameters in the server configuration file described below in accordance with the system. Refer to "A.4 Server Configuration File" for information on how to edit these parameters.

2.11.4.1 Tuning for Abnormality Monitoring of the Operating System or Server

Tuning for abnormal monitoring of the operating system or server depends on the operation when heartbeat abnormality is detected by the heartbeat monitoring of operating systems or servers.

See

Refer to "1.1.1 Monitoring Using Database Multiplexing Mode" for the operation when heartbeat abnormality is detected in the the heartbeat monitoring of operating systems or servers.

2.11.4.1.1 Tuning Abnormality Monitoring for Operations that Use an Arbitration Server for Automatic Degeneration

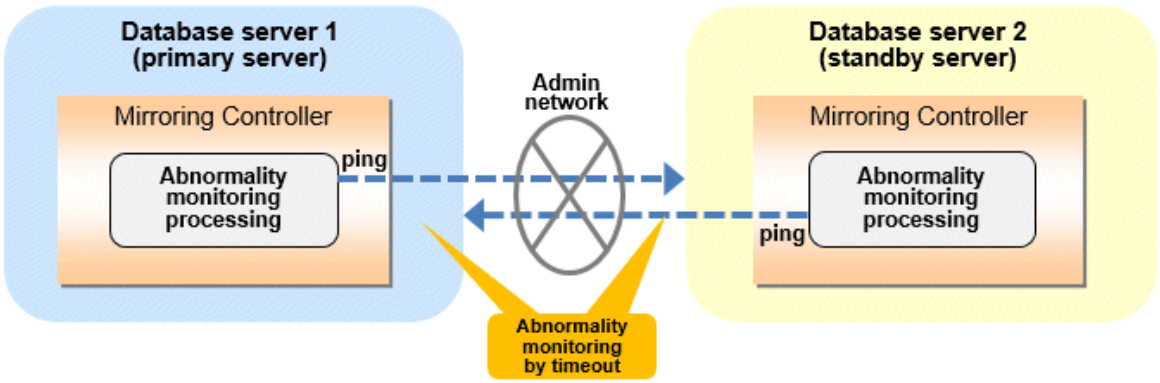
In an operation that use an arbitration server for automatic degeneration, the database server is periodically monitored for abnormalities so that the Mirroring Controller arbitration process can immediately respond to an arbitration request from the Mirroring Controller.The automatic degradation using the arbitration server can optimize the time from error detection to automatic degradation of the operating systems or servers by editing the following parameters.

- Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server
- Parameters for the abnormality monitoring of the operating system or server in the arbitration configuration file
- Parameters for the arbitration processing and fencing

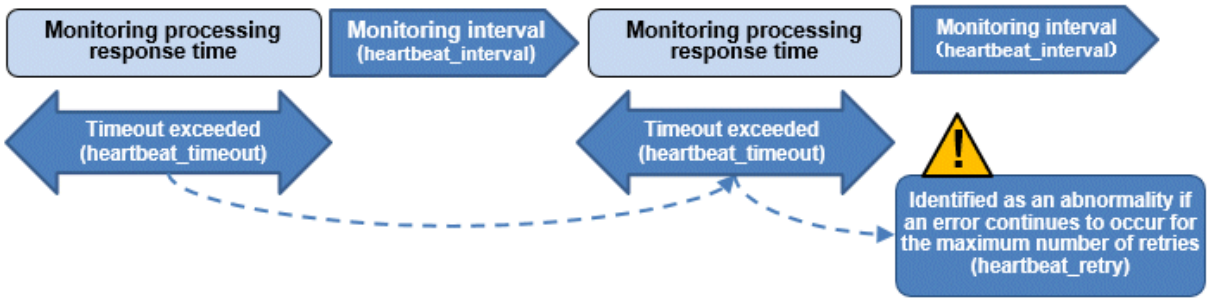
Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server

Table 2.10 Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server

Parameter	Description
Abnormality monitoring interval (heartbeat_interval)	Mirroring Controller is configured so that abnormality monitoring does not place a load on the system. This parameter does not normally need to be set. (The default is 800 milliseconds.)
Abnormality monitoring timeout (heartbeat_timeout)	Take into account the time during which a load is placed continuously on the server or admin network performance. For example, it is envisaged that this parameter will be used in situations such as when performing high-load batch jobs or when a large number of online jobs occur continuously and concurrently. (The default is 1 second.)
Abnormality monitoring retries (heartbeat_retry)	This parameter can be set when needing a safety value for situations in which the value specified for heartbeat_timeout is exceeded, for example, when using systems with fluctuating loads, however, this parameter does not normally need to be set. (The default is 2 times.)



Flow of abnormality monitoring by timeout



The expression for calculating the time required to detect an abnormality by Mirroring Controller is shown below.

```
Abnormality detection time of Mirroring Controller = ( heartbeat_timeout(seconds) +
heartbeat_interval(milliseconds) / 1000 ) x ( heartbeat_retry(number of times) + 1)
```

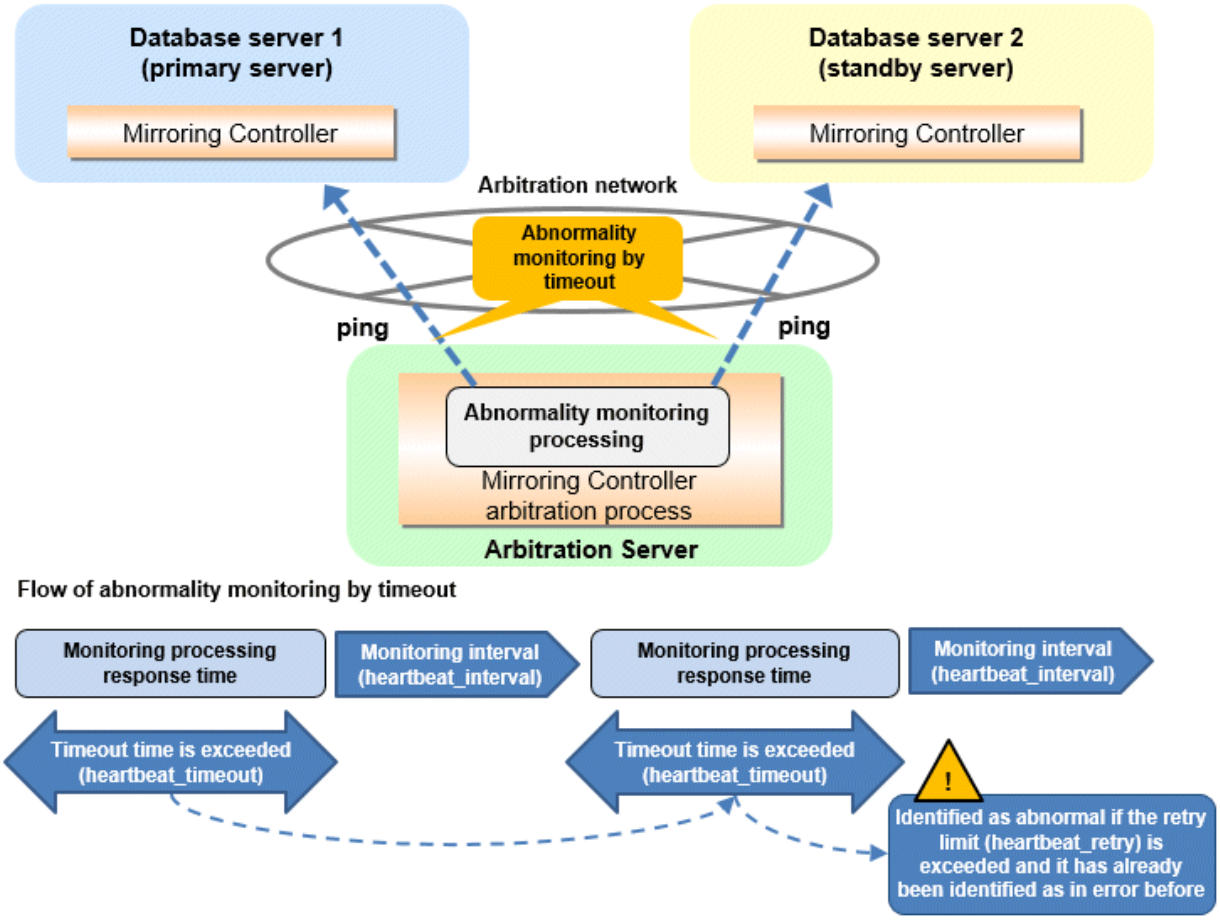
The abnormality detection time when the default value is used is shown below.

```
Abnormality detection time of Mirroring Controller = ( 1 + 800 / 1000 ) x ( 2 + 1 )
            = 5.4(seconds)
```

Parameters for the abnormality monitoring of the operating system or server in the arbitration configuration file

Table 2.11 Parameters for the abnormality monitoring of the operating system or server in the arbitration configuration file

Parameter	Description
Abnormality monitoring interval (heartbeat_interval)	Mirroring Controller arbitration process is configured so that abnormality monitoring does not place a load on the system. This parameter does not normally need to be set. (The default is the value set in heartbeat_interval in the server configuration file of the database server.) (milliseconds).
Abnormality monitoring timeout (heartbeat_timeout)	Take into account the time during which a load is placed continuously on the server and arbitration network capabilities. (The default is the value set in heartbeat_timeout in the server configuration file of the database server.) (seconds).
Abnormality monitoring retries (heartbeat_retry)	This parameter can be set when needing a safety value for situations in which the value specified for heartbeat_timeout is exceeded, for example, when using systems with fluctuating loads, however, this parameter does not normally need to be set. (The default is the value set in heartbeat_retry in the server configuration file of the database server.) (number of times)



The expression for calculating the time required to detect an abnormality by Mirroring Controller arbitration process is shown below.

```
Abnormality detection time of Mirroring Controller arbitration process = ( heartbeat_timeout(seconds)
+ heartbeat_interval(milliseconds) / 1000 ) x ( heartbeat_retry(number of times) + 1)
```

The abnormality detection time when the default value is used is shown below.

```
Abnormality detection time of Mirroring Controller arbitration process = ( 1 + 800 / 1000 ) x ( 2 + 1 )
            = 5.4(seconds)
```

Point

The abnormality detection time of the operation for automatic degradation using the arbitration server can be calculated as follows.

```
Abnormality detection time = Max(Abnormality detection time by Mirroring Controller, Abnormality
detection time by Mirroring Controller arbitration process)
```

Note

If the heartbeat_interval is set in the arbitration configuration file, the relationship between the parameter for operating system or server abnormality monitoring specified in the server configuration file of the database server file and the heartbeat_interval of the arbitration configuration file must satisfy the following relational expression.

```
 Heartbeat_interval in the arbitration configuration file (milliseconds) / 1000 ) <
( heartbeat_timeout(seconds) + heartbeat_interval(milliseconds) / 1000 ) * heartbeat_retry(number of
times) + heartbeat_timeout(seconds)
```

Parameters for the arbitration processing and fencing

Table 2.12 Parameters for the arbitration processing and fencing

Parameter	Description
Arbitration processing timeout (arbitration_timeout in the server configuration file of the database server)	Take into account the time to perform arbitration processing on the Mirroring Controller arbitration process. The value must be greater than or equal to abnormality detection time of Mirroring Controller arbitration process + fencing_command_timeout in the arbitration configuration file (seconds).
Fencing timeout (fencing_command_timeout in the arbitration configuration file)	Take into account the time to execute the fencing command (seconds).

Flow from the abnormality detection to the automatic degeneracy

When performing automatic degradation using the arbitration server, the flow from the abnormality detection in the operating system or server to the occurrence of automatic degeneracy and the parameters is shown below.

<table>
<tr><th>Flow from the abnormality detection to the automatic degeneracy</th><th>Description</th><th colspan="2">Parameter</th></tr>
<tr><td>(1) Abnormality detection</td><td>Mirroring Controller detect the database server operating system or server errors.</td><td colspan="2">Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server</td></tr>
<tr><td>(2) Arbitration request</td><td>Mirroring Controller that detect the operating system or server error asks the Arbitration Server to check the status of the other server's operating system or server.</td><td>-</td><td rowspan="3">arbitration_timeout in the server configuration file of the database server</td></tr>
<tr><td>(3) Arbitration processing</td><td>The Mirroring Controller arbitration process checks the status of the other server's operating system or server. However, if the result of the operating system or server abnormality monitoring by the arbitration server has been determined before the arbitration request from the Mirroring Controller of the database server, this process is not performed.</td><td>Parameters for the abnormality monitoring of the operating system or server in the arbitration configuration file</td></tr>
<tr><td>(4) Fencing</td><td>If the Mirroring Controller arbitration process determines that the other server is an abnomaly of the operating system or server, it fences the other server and isolates it from the cluster system. If the Mirroring Controller arbitration process determines that the operating system or server status is normal, this process and the (6) are not performed.</td><td>fencing_command_timeout in the arbitration configuration file</td></tr>
<tr><td>(5) Return of the arbitration results</td><td>Returns the results of the arbitration to the Mirroring Controller of the database server that requested the arbitration.</td><td colspan="2">-</td></tr>
</table>

Flow from the abnormality detection to the automatic degeneracy	Description	Parameter
(6) Automatic degradation	The automatic degradation is performed. If fencing fails in (4), this procedure is not performed.	-

-: No associated parameters

Note

If the fencing_command parameter is specified in the server configuration file of the database server, the fencing command is invoked on the database server if fencing is successful on the arbitration server. In that case, add the value of the fencing_command_timeout parameter in the server configuration file of the database server to the estimate.

Figure 2.1 When the Mirroring Controller on the primary server detects an operating system or server error

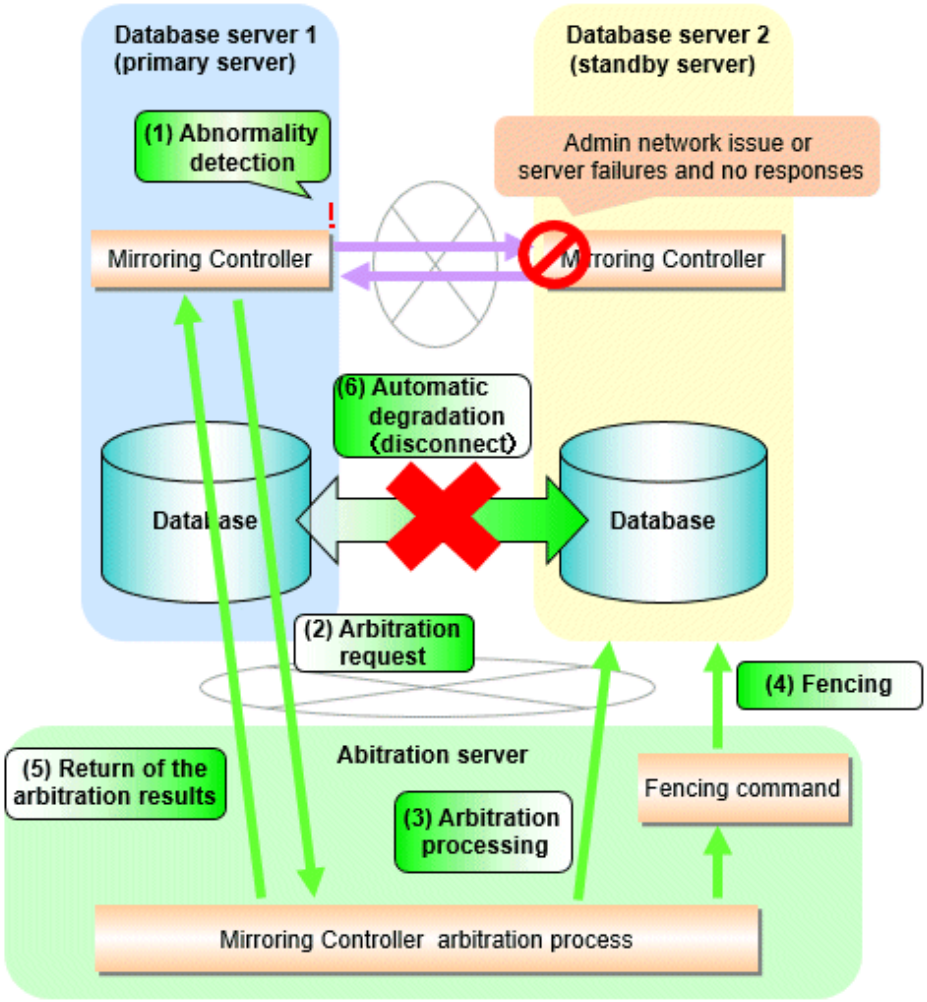
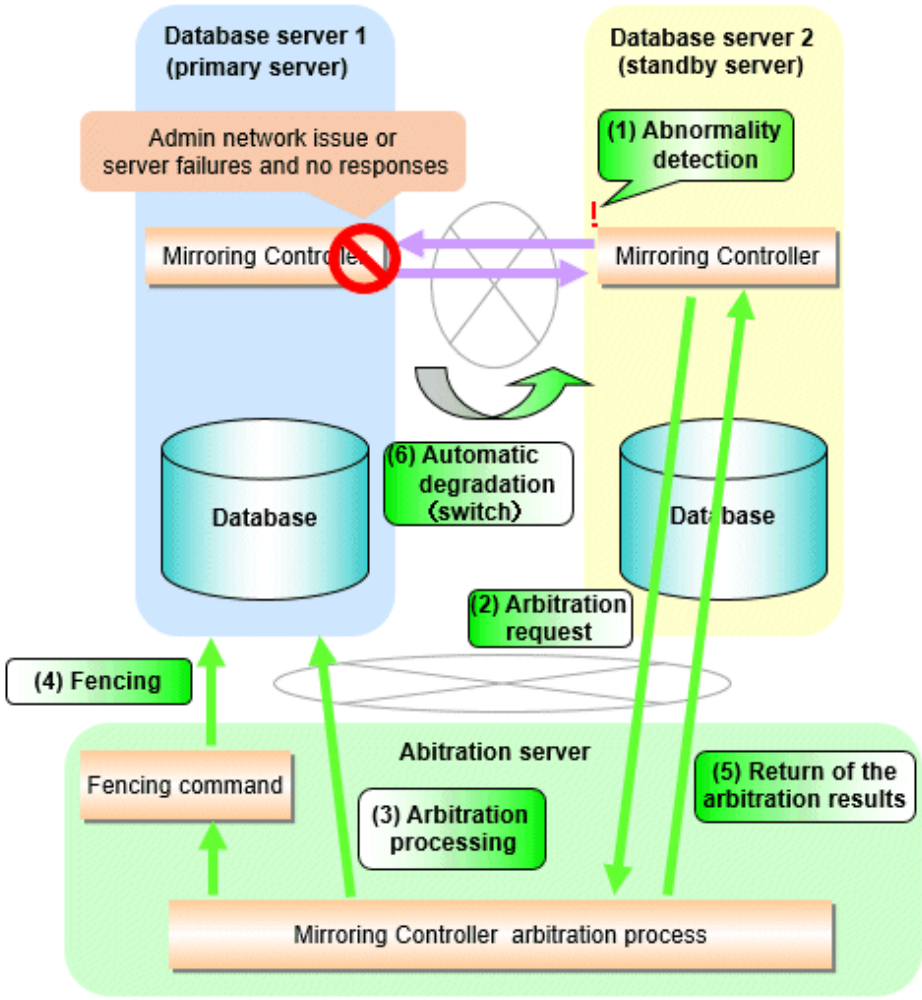


Figure 2.2 When the Mirroring Controller on the standby server detects an operating system or server error



2.11.4.1.2 Tuning Abnormality Monitoring for Operations that Perform Automatic Degeneration by Calling a User Exit that Determines Degeneration

In an operation that perform automatic degeneration by calling a user exit that determines degeneration, you can optimize the time from operating system or server abnormality detection to automatic degradation by editing the operating system or server abnormality monitoring parameters and parameters related to arbitration processing and fencing in the server configuration file of the database server. Refer to "Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server" for information on the operating system or server abnormality monitoring parameters in the server configuration file of the database server.

Table 2.13 Parameters for the arbitration processing and fencing

Parameter	Description
Arbitration processing timeout (arbitration_command_timeout)	Take into account the time to execute the arbitration command(seconds).
Fencing timeout (fencing_command_timeout)	Take into account the time to execute the fencing command (seconds).

Flow from the abnormality detection to the automatic degeneracy

When performing automatic degradation by calling a user exit that determines degeneration, the flow from the abnormality detection in the operating system or server to the occurrence of automatic degeneracy and the parameters is shown below.

Flow from the abnormality detection to the automatic degeneracy	Description	Parameter
(1) Abnormality detection	Mirroring Controller detect the database server operating system or server errors.	Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server
(2) Arbitration processing	An arbitration command is executed to check the status of the other server's operating system or server.	arbitration_command_timeout in the server configuration file of the database server
(3) Fencing	If the operating system or server status of the other server is abnomal in (2), it fences the other server and isolates it from the cluster system. If the operating system or server status of the other server is nomal in (2), this process and (4) are not executed.	fencing_command_timeout in the server configuration file of the database server
(4) Automatic degradation	The automatic degradation is performed. If fencing fails in (3), this procedure is not performed.	-

-: No associated parameters

Figure 2.3 When the Mirroring Controller on the primary server detects an operating system or server error

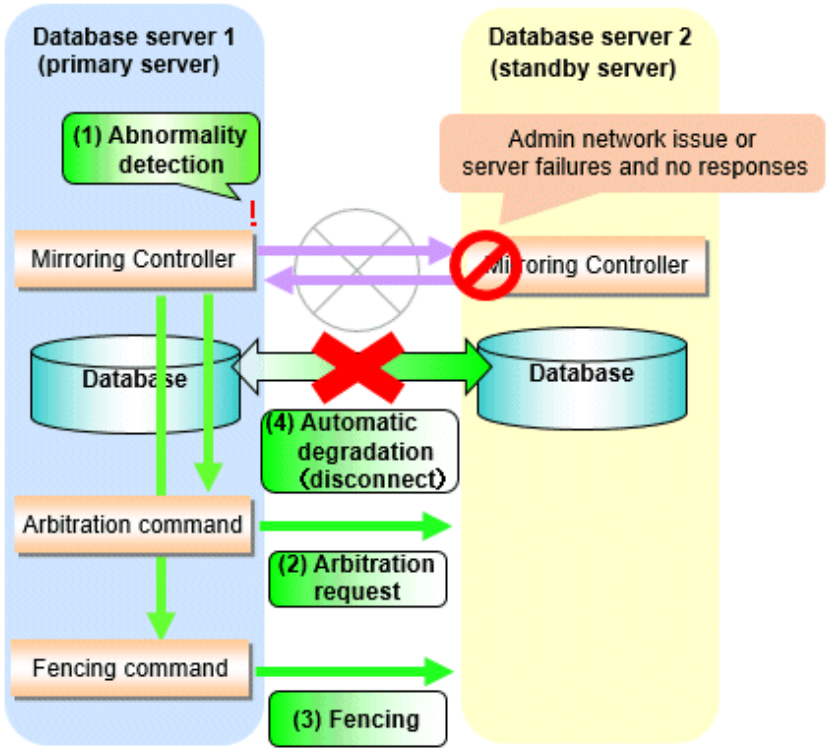
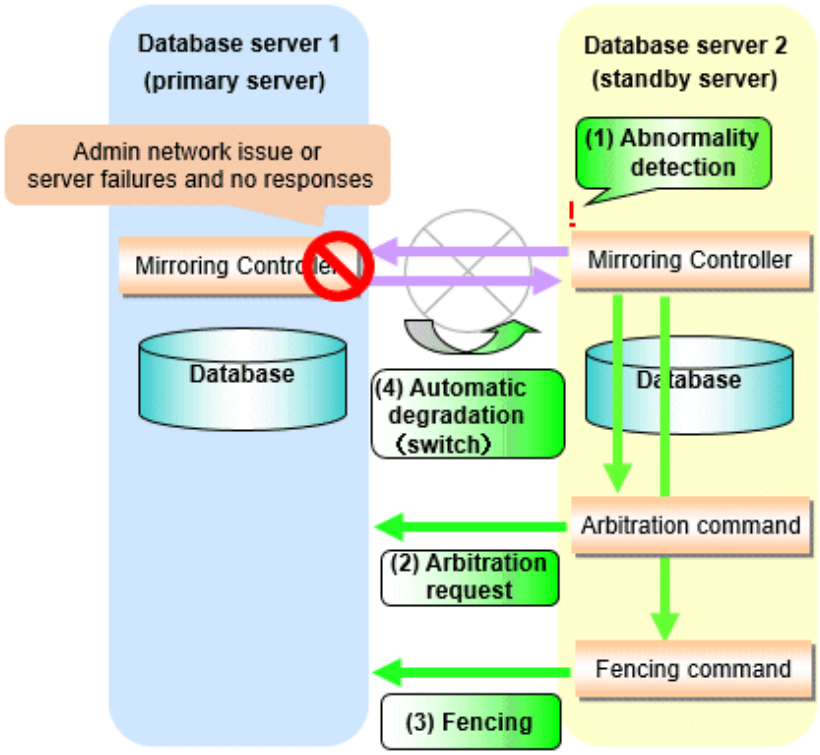


Figure 2.4 When the Mirroring Controller on the standby server detects an operating system or server error



2.11.4.1.3 Tuning Abnormality Monitoring for Operations that Notify Messages

In an operation that notify messages, you can optimize the abnormality detection time by editing the operating system or server abnormality monitoring parameters in the server configuration file of the database server. Refer to "Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server" for information on the operating system or server abnormality monitoring parameters in the server configuration file of the database server. In addition, when the Mirroring Controller detects an error, it does not perform the arbitration processing, fencing, or automatic degradation, but only notification messages is performed.

2.11.4.1.4 Tuning Abnormality Monitoring for Operations that Perform Automatic Degenerate Unconditionally due to Heartbeat Abnormality

In an operation that perform automatic degenerate unconditionally due to heartbeat abnormality, you can optimize the time from operating system or server abnormality detection to automatic degradation by editing the operating system or server abnormality monitoring parameters in the server configuration file of the database server. Refer to "Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server" for information on the operating system or server abnormality monitoring parameters in the server configuration file of the database server. In addition, when the Mirroring Controller detects an error, it does not perform the arbitration processing, fencing, or automatic degradation, but only automatic degenerate unconditionally is performed.

Note

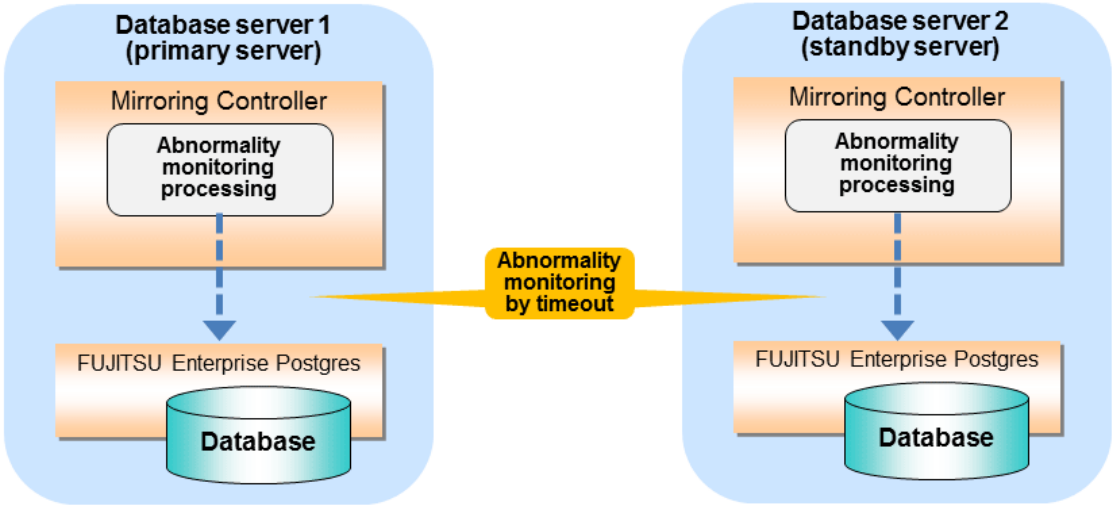
Refer to "Appendix D Notes on Performing Automatic Degradation Immediately after a Heartbeat Abnormality" for notes on the operation that perform automatic degenerate unconditionally due to heartbeat abnormality.

2.11.4.2 Tuning for Abnormality Monitoring of Darabase Processes

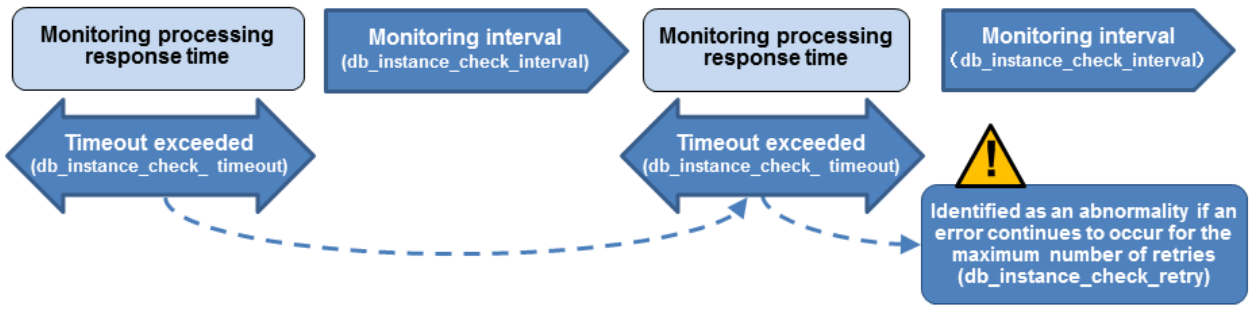
In an abnormality monitoring of database pocesses, you can optimize by editing the following parameters in the server configuration file of the database server.

Table 2.14 Parameters for abnormality monitoring of database processes

Parameter	Description
Abnormality monitoring interval (db_instance_check_interval)	Abnormality monitoring by Mirroring Controller is set so as not to place load on the system, but normally it does not need to be set. (The default is the value set in heartbeat_interval.) (milliseconds)
Timeout for abnormality monitoring of database processes (db_instance_check_timeout)	Take into account the time during which a load is placed continuously on the database. For example, it is envisaged that this parameter will be used in situations such as when performing high-load batch jobs or when a large number of online jobs occur continuously and concurrently. (The default is the value set in heartbeat_timeout.) (seconds)
Abnormality monitoring retries (db_instance_check_retry)	This parameter can be set when needing a safety value for situations in which the value specified for db_instance_check_timeout is exceeded, for example, when using systems with fluctuating loads, however, this parameter does not normally need to be set. (The default is the value set in heartbeat_retry.) (number of times)



Flow of abnormality monitoring by timeout



The expression for calculating the time required to detect an abnormality is shown below.

```
Abnormality detection time = ( db_instance_check_timeout(seconds) +
db_instance_check_interval(milliseconds) / 1000 ) x ( db_instance_check_retry(number of times) + 1 )
```

The abnormality detection time when the default value is used is shown below.

```
Abnormality detection time = ( 1 + 800 / 1000 ) x ( 2 + 1 )
             = 5.4(seconds)
```

Note

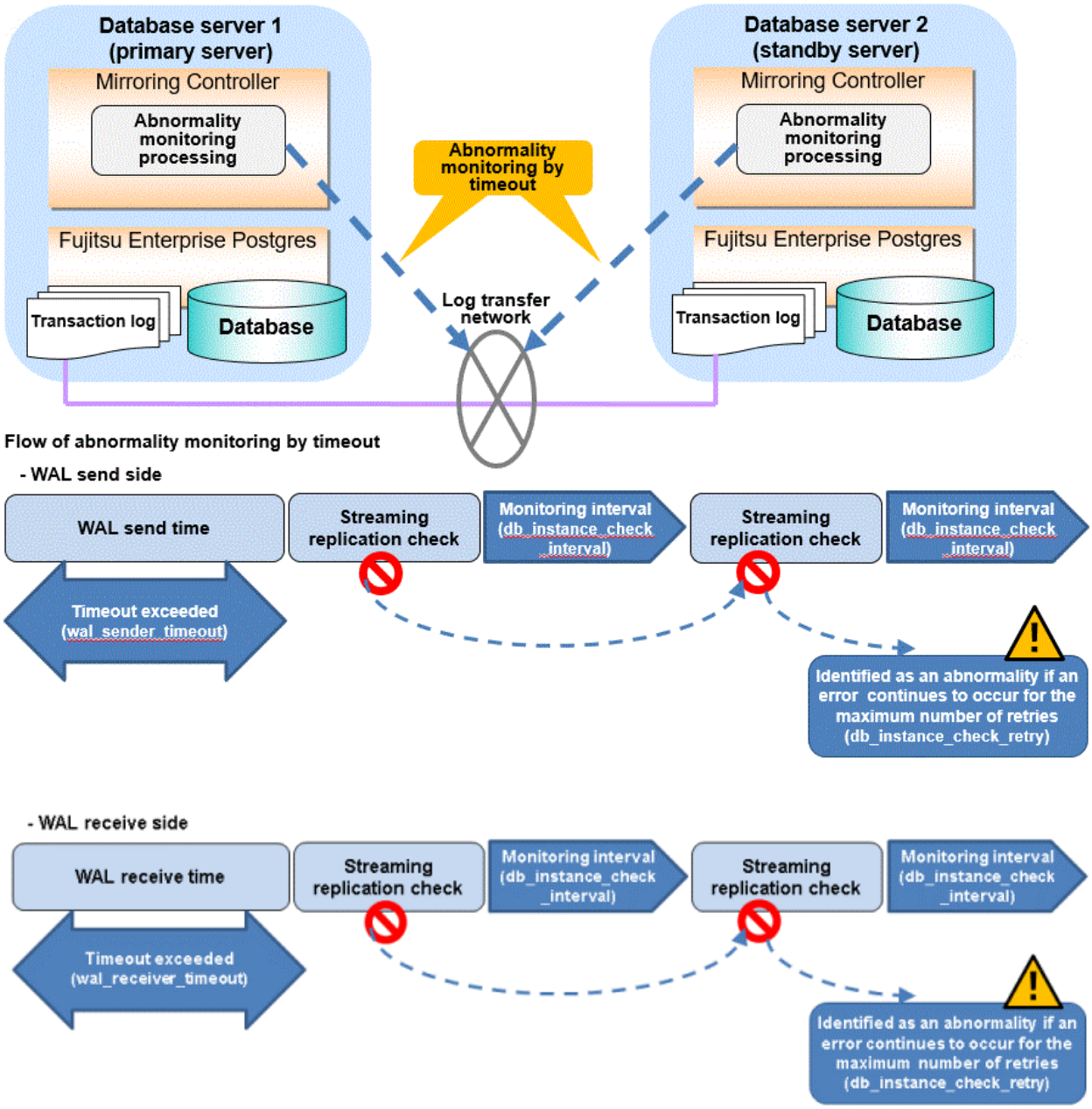
- If the db_instance_timeout_action parameter in *serverIdentifier*.conf is set to "message", and the db_instance_check_timeout parameter is set to a short value, a crash of the database process will be detected as "no response", and it may take time for automatic degradation to occur. Therefore, specify an appropriate timeout for db_instance_check_timeout.
- If a high load on the database and an event that prevents connection to an instance occur at the same time, it is judged as abnormal without retrying monitoring.

2.11.4.3 Tuning for Abnormality Monitoring of Streaming Replication

In an abnormality monitoring of streaming replication, you can optimize by editing the following parameters in the server configuration file of the database server.

Table 2.15 Parameters for abnormality monitoring of streaming replication

Parameter	Description
Abnormality monitoring interval (db_instance_check_interval)	Abnormality monitoring by Mirroring Controller is set so as not to place load on the system, but normally it does not need to be set. (The default is the value set in heartbeat_interval.) (milliseconds)
Abnormality monitoring retries (db_instance_check_retry)	This parameter can be set when needing a safety value, such as when it is anticipated that a temporary log transfer LAN error may occur, but it does not normally need to be set. (The default is the value set in heartbeat_retry.) (number of times)
Timeout for abnormality monitoring of streaming replication (wal_sender_timeout and wal_receiver_timeout in postgresql.conf)	Take into account the capacity and load of the log transfer network and the time during which a load is placed continuously on the database. For example, if there is a succession of data update jobs that generate a high WAL volume, you must configure the settings to avoid misdetection. (The default is 60 seconds.)



The expression for calculating the time required to detect an abnormality is shown below.

```
Abnormality detection time = ( wal_sender_timeout(seconds) +
db_instance_check_interval(milliseconds) / 1000 x ( disk_check_retry(number of times) + 1 ) )  Or,
            = ( wal_receiver_timeout(seconds) + db_instance_check_interval(milliseconds) / 1000 x
( disk_check_retry(number of times) + 1 ) )
```

The abnormality detection time when the default value is used is shown below.

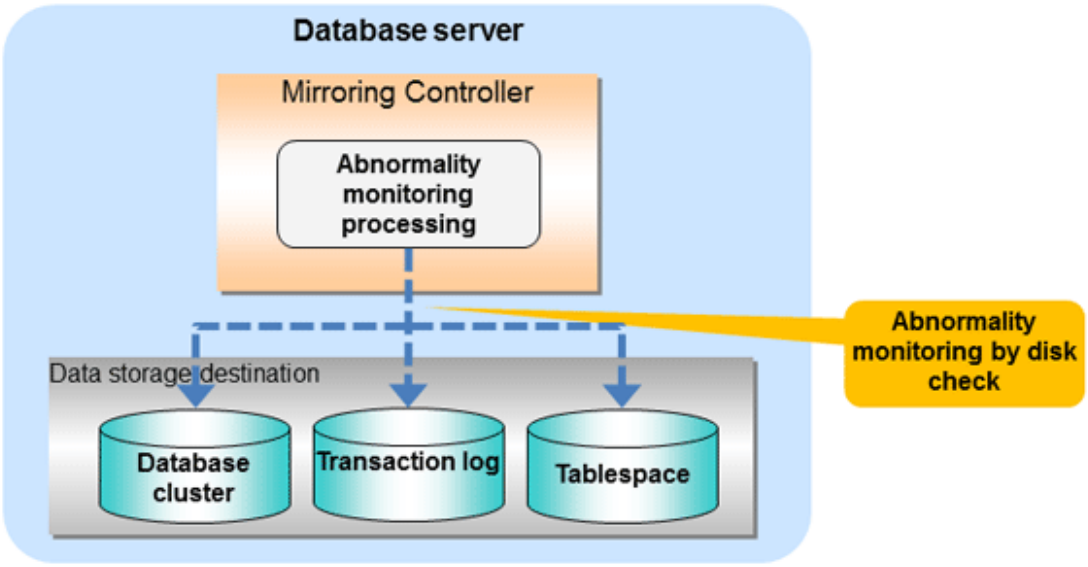
```
Abnormality detection time = 60 + (800 / 1000 x (2 + 1))
            = 62.4(seconds)
```

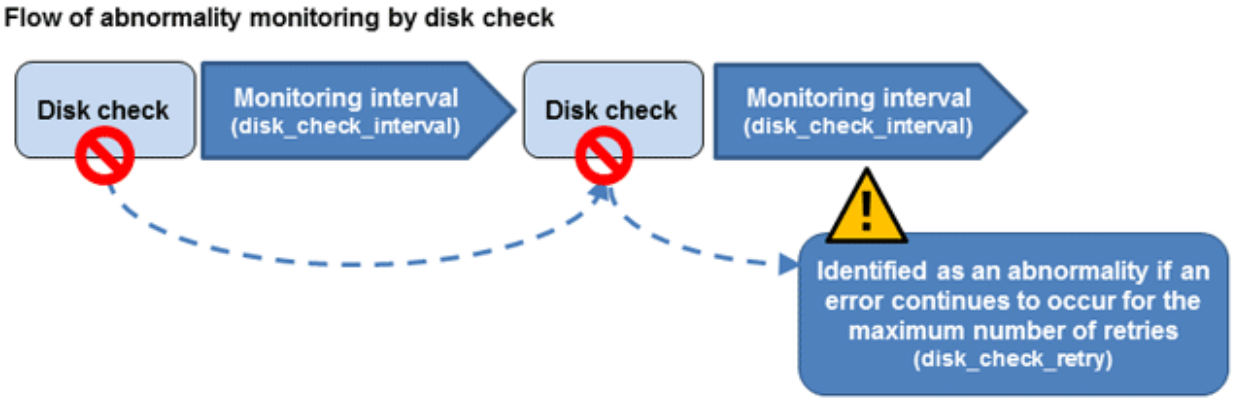
2.11.4.4 Tuning for Disk Abnormality Monitoring

In an abnormality monitoring of the disk, you can optimize by editing the following parameters in the server configuration file of the database server.

Table 2.16 Parameters for disk abnormality monitoring

Parameter	Description
Abnormality monitoring interval (disk_check_interval)	Abnormality monitoring by Mirroring Controller is set so as not to place load on the system, but normally it does not need to be set. (The default is the value set in heartbeat_interval.) (milliseconds)
Abnormality monitoring retries (disk_check_retry)	This parameter can be set when needing a safety value, such as when it is anticipated that a temporary disk input/output error may occur, but normally it does not need to be set. (The default is the value set in heartbeat_retry.) (number of times)





The expression for calculating the time required to detect an abnormality is shown below.

```
Abnormality detection time = disk_check_interval (milliseconds) / 1000 x ( disk_check_retry(number of times) + 1 )
```

The abnormality detection time when the default value is used is shown below.

```
Abnormality detection time = 800 / 1000  x ( 2 + 1 )
            = 2.4(seconds)
```

Note

- The tuning described above impacts on the time taken from detection of a timeout until switching the primary server. Therefore, modify the values while taking into account the switch/disconnection time, using a design for which misdetection does not occur.
- Immediately selecting automatic degradation when a heartbeat abnormality occurs in operating system or server heartbeat monitoring risks causing split brain. Refer to "Appendix D Notes on Performing Automatic Degradation Immediately after a Heartbeat Abnormality" for details.

Information

Mirroring Controller uses connections to database instances and SQL access to monitor abnormality in some resources targeted for monitoring. The connection destination database names and connection user names used for abnormality monitoring conform to the parameters in the server configuration file. The application name is "mc_agent".

2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances

Multiplexed instances and Mirroring Controller can be started and stopped automatically in line with the starting and stopping of the operating system of the database server.

Note

To guarantee the startup sequence of Mirroring Controller on the primary and standby servers, first confirm that the primary server has started, and then start the standby servers in sequence.

The startup sequence of the Mirroring Controller process on the database server and the Mirroring Controller arbitration process on the arbitration server is not guaranteed. If the arbitration server cannot be started first, execute the mc_ctl command in start mode with the --async-connect-arbiter option specified to start the Mirroring Controller process.

If you start the Mirroring Controller and multiplexed instances, wait for time correction, network setup, and so on.

Perform the following procedure:

1. Create a unit file

   Copy the unit file sample stored in the directory below, and revise it to match the target instance.

   Sample file

   ```
   /installDir/share/mcoi.service.sample
   ```

   Example)

   In the following example, the installation directory is "/opt/fsepv<x>server64", and the instance name is "inst1". Note that "<x>" indicates the product version.

   ```
   # cp /opt/fsepv<x>server64/share/mcoi.service.sample /usr/lib/systemd/system/mcoi_inst1.service
   ```

   Revise the underlined portions of the options below in the unit file.

Section	Option	Specified value	Description
Unit	Description	Fujitsu Enterprise Postgres MirroringController *instanceName*	Specifies the feature overview. Specifies the name of the target instance. (*1)
Service	ExecStart	/bin/bash -c '*installDir*/bin/mc_std start *installDir*	Command to be executed when the service is started.

Section	Option	Specified value	Description
		MirroringControllerManagementDir' *mc_ctlOption*'	Specify the option you want to add when the mc_ctl command is executed without the -M option in the mc_ctl option. Note that the content specified in this mc_ctl option is carried over from the mc_std command to the mc_ctl command. (*2)
	ExecStop	/bin/bash -c '*installDir*/bin/mc_std stop *installDir* *MirroringControllerManagementDir*' *mc_ctlOption*'	Command to be executed when the service is stopped. Specify the option you want to add when the mc_ctl command is executed without the -M option in the mc_ctl option. However, to use the --mc-only option to stop only Mirroring Controller, you must use the --mc-only option at startup. Note that the content specified in this mc_ctl option is carried over from the mc_std command to the mc_ctl command.
	User	User	OS user account of the instance administrator user.
	Group	Group	Group to which the instance administrator user belongs.

*1: The instance name should be as nameThatIdentifiesTheInstance.
The naming conventions for identifying the instance are as follows:

- Up to 16 bytes
- The first character must be an ASCII alphabetic character
- The other characters must be ASCII alphanumeric characters

*2: When the arbitration server is used for automatic degradation, start the Mirroring Controller arbitration process on the arbitration server and then start the Mirroring Controller process on the database server. If the arbitration server cannot be started first, specify the --async-connect-arbiter option to start the Mirroring Controller process.

2. Enable automatic start and stop

   As the OS superuser, use the systemctl command to enable automatic start and stop.

   Example)

```
# systemctl enable mcoi_inst1.service
```

Note

If automatic start and stop of Mirroring Controller has been configured, to stop Mirroring Controller, do not use the mc_ctl command, but instead use the systemctl command as the OS superuser.

Example)

```
# systemctl stop mcoi_inst1.service
```

If the instance does not stop, refer to "Actions in Response to Failure to Stop an Instance" in the Operation Guide to stop the instance. Then, specify the -e option in the mc_ctl command to forcibly stop Mirroring Controller.

Example)

```
$ mc_ctl stop -M /mcdir/inst1 -e
```

If Mirroring Controller is stopped using the mc_ctl command, the message below is output to the system log, however there is no issue because automatic stop is executed by systemd.

Message

```
FATAL: failed to stop Mirroring Controller target server:"{0}" (MCA00043)
```

2.13 Setting Automatic Start and Stop of the Mirroring Controller Arbitration Process

You can automatically start or stop the Mirroring Controller arbitration process when the operating system on the arbitration server is started or stopped.

Linux

Note

If you start the Mirroring Controller arbitration process, wait for time correction, network setup, and so on.

Perform the following procedure:

1. Create a unit file.

   Copy the unit file sample stored in the directory below, and revise it to match the target instance.

   Sample file

   ```
   /installDir/share/mcarboi.service.sample
   ```

   Example)
   In the following example, the installation directory is "/opt/fsepv<x>assistant", and the identifier of the arbitration process is "arbiter1". Note that "<x>" indicates the product version.

   ```
   # cp /opt/fsepv<x>assistant/share/mcarboi.service.sample /usr/lib/systemd/system/
   mcarboi_arbiter1.service
   ```

   Revise the underlined portions of the options below in the unit file.

Section	Option	Specified value	Description
Unit	Description	Fujitsu Enterprise Postgres Mirroring Controller Arbiter <*arbitrationProcessId*>	Specifies the feature overview. Specifies the identifier of the targeted arbitration process. (*1)
Service	ExecStart	/bin/bash -c '*installDir*/bin/mc_arb_std start *installDir mirroringControllerArbitrationProcessMgmtDir mc_arbOption*'	Command to be executed when the service is started. Specify the option you want to add when the mc_arb command is executed without the -M option in the mc_arb option.

Section	Option	Specified value	Description
			Note that the content specified in this mc_arb option is carried over from the mc_arb_std command in "Specified value" to the mc_arb command.
	ExecStop	/bin/bash -c '*installDir*/bin/ mc_arb_std stop *installDir mirroringControllerArbitrationProcessMgmtDir mc_arbOption*'	Command to be executed when the service is stopped. Specify the option you want to add when the mc_arb command is executed without the -M option in the mc_arb option. Note that the content specified in this mc_arb option is carried over from the mc_arb_std command in "Specified value" to the mc_arb command.
	User	User	Specify the account of the operating system user.
	Group	Group	Specify the group to which the user belongs.

*1: The arbitration process identifier used here is a name for identifying the Mirroring Controller arbitration process.

The naming conventions for identifying the Mirroring Controller arbitration process are as follows:

- Up to 16 bytes
- The first character must be an ASCII alphabetic character
- The other characters must be ASCII alphanumeric characters

2. Enable automatic start and stop.

   As the operating system superuser, use the systemctl command to enable automatic start and stop.

   Example)

```
# systemctl enable mcarboi_arbiter1.service
```

Windows

You can configure the Windows service to perform automatic start and stop.

Configuring during setup

When registering the Mirroring Controller arbitration process as a Windows service in "2.3.1 Configuring the Arbitration Server", specify "auto" for the -S option of the register mode used with the mc_arb command.

Example)

```
> mc_arb register -M D:\mcdir\inst1 -P ******** -S auto
```

Changing the configuration after setup

Use the sc config command to change the configuration of the Windows service of the Mirroring Controller arbitration process.

Example)
The configuration of the registered service name "Mirroring_Controller_Arbiter1" is changed.

```
> sc config "Mirroring_Controller_Arbiter1" start= auto
```

See

Refer to sc command help for information on how to configure the service.

Information

You can check the registration status in the Windows service window or by using the sc qc command.

2.14 Backup Operation

This section explains the backup operation for database multiplexing mode.

2.14.1 Backing up Database Multiplexing Mode Information

When changing the Mirroring Controller settings, in addition to backing up the database, back up the configuration file in the Mirroring Controller management directory so that the Mirroring Controller settings are not lost.

When the arbitration server is used for automatic degradation, also back up the configuration file in the Mirroring Controller arbitration process management directory.

2.14.2 Database Backup Operation

Using database multiplexing mode is the same as obtaining the backup data on the standby server as a safeguard against a disk failure. Note that all server disks may be corrupted due to some cause.

As a safeguard against this type of case, execute the pgx_dmpall command on the primary server to create the backup data.

However, it is not definite as to which server runs as the primary server, so ensure that the pgx_dmpall command is executed periodically on all servers, so that the backup data will be obtained. For example, create a script to obtain the backup data, and set it in the operation management software.

Point

When the pgx_dmpall command is executed on the standby server, it will not match the statuses, however the error message shown below will be output and return the value "1".

If a script that ignores only this type of error is executed on all servers, the backup data of the primary server can be obtained.

Error message

```
ERROR:recovery is in progress (10095)
```

Note

- Consider the possibility that the server that runs as the primary server may be destroyed alongside the backup data, so it is recommended to promote another server to become the primary server, and then back up the data on the new primary server without waiting for the next scheduled backup.
- Specify the same backup directory name for the primary and standby servers. If different backup directory names are specified, and recovery is performed using the backup data of the other server, the recovery cannot be performed correctly.

See

- Period backups allow shorter recovery time and reduction in disk usage. Refer to "Backing Up the Database" in the Operation Guide for details on the backup operation.
- Refer to "Chapter 4 Action Required when an Error Occurs in Database Multiplexing Mode" for details on recovery based on the backup data that was obtained using the pgx_dmpall command.

Chapter 3 Operations in Database Multiplexing Mode

This chapter describes the periodic operations that are performed when running database multiplexing mode.

The periodic operations are the same as the operations on a single server.

See

Refer to "Periodic Operations" in the Operation Guide for information on the periodic operations.

3.1 Starting and Stopping the Mirroring Controller Arbitration Process

This section describes how to start and stop the Mirroring Controller arbitration process.

3.1.1 Starting the Mirroring Controller Arbitration Process

Linux

While the Mirroring Controller arbitration process is in a stopped state, execute the mc_arb command in start mode to start the Mirroring Controller arbitration process.

Example)

```
$ mc_arb start -M /mcarb_dir/arbiter1
```

Windows

The Mirroring Controller arbitration process can be started using one of the following options:

- Using the mc_arb command
- Starting the service on system startup

Using the mc_arb command

While the Mirroring Controller arbitration process is in a stopped state, execute the mc_arb command from the command prompt to start the Mirroring Controller arbitration process.

Example)

```
> mc_arb start -M D:\mcarb_dir\arbiter1
```

Starting the service on system startup

Specify automatic start when registering the Mirroring Controller arbitration process to the Windows service during setup of database multiplexing mode. Accordingly, the Mirroring Controller arbitration process service will start on startup of the operating system.

See

Refer to the Reference for information on how to specify the mc_arb command.

3.1.2 Stopping the Mirroring Controller Arbitration Process

Linux

While the Mirroring Controller arbitration process is running, execute the mc_arb command in stop mode to stop the Mirroring Controller arbitration process.

Example)

```
$ mc_arb stop -M /mcarb_dir/arbiter1
```

Windows

The Mirroring Controller arbitration process can be stopped using one of the following options:

- Using the mc_arb command
- Stopping the service

Using the mc_arb command

While the Mirroring Controller arbitration process is running, execute the mc_arb command in stop mode from the command prompt to stop the Mirroring Controller arbitration process.

Example)

```
> mc_arb stop -M D:\mcarb_dir\arbiter1
```

Stopping the service

Select [Administrative Tools], then [Services] to open the [Services] window, and then select the Mirroring Controller service and click the [Stop] menu.

See

Refer to the Reference for information on how to specify the mc_arb command.

Note

- The arbitration server will be forcibly stopped when the service is stopped.
- Before shutting down the operating system on the arbitration server, either stop the Mirroring Controller on the primary server or standby server or shut down the operating system on the primary server or standby server.

3.2 Starting and Stopping Mirroring Controller

When database multiplexing mode is used, use the mc_ctl command to start and stop the instance and Mirroring Controller at the same time.

Do not start or stop the instance by itself.

Starting Mirroring Controller

While Mirroring Controller is in a stopped state, execute the mc_ctl command in start mode to start Mirroring Controller.

Enabling automatic switch/disconnection

Execute the mc_ctl command in start mode.

Example)

```
$ mc_ctl start -M /mcdir/inst1
```

When only the instance is started and stopped, the following will happen:

- When only the instance is started

  Features such as automatic switch and automatic disconnection will not work until Mirroring Controller is started.

- When only the instance is stopped

  Mirroring Controller determines that an error has occurred in the instance, and performs an unnecessary automatic switch.

  Automatic switch may also stop working correctly in some cases.

Disabling automatic switch/disconnection

Execute the mc_ctl command in start mode with the -F option specified.

Example)

```
$ mc_ctl start -M /mcdir/inst1 -F
```

When only the instance is started and stopped, the following will happen:

- When only the instance is started

  Errors indicated in "1.1 What is Database Multiplexing Mode" will not be detected until Mirroring Controller is started.

- When only the instance is stopped

  Mirroring Controller determines that an error has occurred in the instance, and outputs an error to the system log.

Point

- To start the Mirroring Controller process only, execute the mc_ctl command in start mode with the --mc-only option specified.
- After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.
- When the arbitration server is used for automatic degradation, the Mirroring Controller process startup fails on the database server if the Mirroring Controller arbitration process has not been started on the arbitration server in advance. However, even if the Mirroring Controller arbitration process cannot be started in advance, the Mirroring Controller process can be started by specifying the --async-connect-arbiter option in the mc_ctl command.

Note

- When the arbitration server is used for automatic degradation, the database server must connect to the arbitration server, and as a result, Mirroring Controller startup may take longer.
- Mirroring Controller startup usually fails if the standby server is mistakenly started as the primary server or if the old primary server is not recovered after the switch and is then mistakenly started as the primary server. However, if the admin network is disconnected, then startup does not fail, and both servers may become primary servers. Therefore, ensure that the admin network is connected before starting Mirroring Controller.

Stopping Mirroring Controller

While Mirroring Controller is running, execute the mc_ctl command in stop mode to stop Mirroring Controller process.

Example)

```
$ mc_ctl stop -M /mcdir/inst1
```

Point

To stop the Mirroring Controller process only, execute the mc_ctl command in stop mode with the --mc-only option specified.

Note

To prevent an unintended automatic switch, before shutting down the operating system on the primary server, you must stop the Mirroring Controller, or shut down the operating system on the standby server.

See

Refer to the Reference for information on how to specify the mc_ctl command.

3.3 Checking the Database Multiplexing Mode Status

3.3.1 Checking the Status of the Database Server

This section describes how to check the status of the database server.

Check the multiplexed database status by executing the mc_ctl command in status mode.

Additionally, errors can be detected by monitoring the Mirroring Controller messages. If the status or messages are monitored periodically, you can react quickly following an automatic switch failure.

Checking the status of the multiplexing database

When the mc_ctl command is executed, the details of the multiplexing configuration, information about whether switch is possible following the error, and location and details of the error that caused the switch or disconnection are displayed.

After starting database multiplexing mode, execute the mc_ctl command in status mode to check the multiplexing status.

An example of the status displayed when the mc_ctl command is executed is shown below.

Example)

```
$ mc_ctl status -M /mcdir/inst1

mirroring status
----------------
switchable
server_id  host_role       host            host_status  db_proc_status       disk_status
-------------------------------------------------------------------------------------------------
----
server1    primary         192.0.2.100     normal       normal                 normal
server2    standby         192.0.2.110     normal       normal                 normal
```

Checking the status of connection to the Mirroring Controller arbitration process

When the arbitration server is used for automatic degradation, the status of the connection to the Mirroring Controller arbitration process can be checked by specifying the --arbiter option. If the output status is "online", it indicates that an arbitration request can be made from the database server to the arbitration server. When the arbitration server is used for automatic degradation, regularly execute the command in status mode with the --arbiter option specified and check that the output status is "online".

Example)
The mc_ctl command is executed with the --arbiter option specified, and the status is output.

```
$ mc_ctl status --arbiter -M /mcdir/inst1
```

```
arbiter_id   host            status
-------------------------------------
arbiter      192.0.3.120     online
```

Checking the status of data synchronization

Additionally, by referencing the pg_stat_replication statistics view on the primary server, the data synchronization status can be confirmed. However, when creating the monitoring program, note that the content of pg_stat_replication may be changed in the future.

The following example shows that the locations of the transaction log after it is sent and received (sent_lsn, replay_lsn) match, and that they are fully synchronized.

Example)

```
postgres=# select * from pg_stat_replication;
-[ RECORD 1 ]----+-----------------------------
pid              | 10651
usesysid         | 10
usename          | fsep
application_name | standby
client_addr      | 192.0.2.210
client_hostname  |
client_port      | 55098
backend_start    | 2022-03-23 11:17:49.628793+09
backend_xmin     |
state            | streaming
sent_lsn         | 0/3000060
write_lsn        | 0/3000060
flush_lsn        | 0/3000060
replay_lsn       | 0/3000060
write_lag        |
flush_lag        |
replay_lag       |
sync_priority    | 1
sync_state       | sync
reply_time       | 2022-03-23 11:23:27.703366+09
```

- Refer to "mc_ctl" in Reference for information on the command.
- Refer to "Notes on Application Compatibility" in the Application Development Guide for information on retaining application compatibility.
- Refer to "The Statistics Collector" in "Server Administration" in the PostgreSQL Documentation for details on pg_stat_replication.

3.3.2 Checking the Status of the Arbitration Server

This section describes how to check the status of the arbitration server.

The status of the connection between the Mirroring Controller arbitration process and primary server/standby server can be checked by executing the mc_arb command in status mode.

The example below executes the mc_arb command, and shows the status.

Linux

Example)

```
$ mc_arb status -M /mcarb_dir/arbiter1

server_id    host            status
--------------------------------------
```

```
server1      192.0.3.100     online
server2      192.0.3.110     online
```

Windows

Example)

```
> mc_arb status -M D:\mcarb_dir\arbiter1

server_id    host            status
-------------------------------------
server1      192.0.3.100     online
server2      192.0.3.110     online
```

3.4 Manually Switching the Primary Server

The primary server cannot be switched automatically in the following case:

- If automatic switch/disconnection is disabled
- If output of messages is selected for heartbeat abnormalities during heartbeat monitoring of the operating system or server and the operating system/server crashes or becomes unresponsive

In this case, to manually switch the primary server, execute the mc_ctl command in switch mode on either the primary server or the standby server.

Example)

```
$ mc_ctl switch -M /mcdir/inst1
```

Point

If automatic switch/disconnection is enabled, it is possible to perform switch of primary server at any time.

3.5 Manually Disconnecting the Standby Server

The procedure to perform disconnection of the standby server differs depending on whether the automatic switch/disconnection is enabled or disabled.

If automatic switch/disconnection is enabled

Execute the mc_ctl command in stop mode on the standby server.

Example)

```
$ mc_ctl stop -M /mcdir/inst1
```

If automatic switch/disconnection is disabled

1. Execute the mc_ctl command in stop mode on the standby server.

   Example)

   ```
   $ mc_ctl stop -M /mcdir/inst1
   ```

2. Comment out the synchronous_standby_names parameter in the postgresql.conf file on the primary server.
3. Execute the pg_ctl command in reload mode on the primary server.

   Example)

   ```
   $ pg_ctl reload -D /database/inst1
   ```

Point

If automatic start and stop of Mirroring Controller has been configured using systemd, do not use the mc_ctl command, but instead use the systemctl command. Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for details.

3.6 Action Required when a Heartbeat Abnormality is Detected

The message below is output when a heartbeat abnormality is detected during heartbeat monitoring of operating systems or servers:

```
detected an error on the monitored object "server(server identifier name)": no response:ping timeout
(MCA00019)
```

If the heartbeat_error_action parameter in *serverIdentifier*.conf is set to "message", even if automatic switch/disconnection is enabled and Mirroring Controller is started, automatic switch/disconnection is not performed when a heartbeat abnormality is detected. Therefore, user action will be necessary.

This section explains the action required when the heartbeat_error_action parameter is set to "message" and a heartbeat abnormality is detected.

1. Identify the cause of the heartbeat abnormality. The possible causes are below:
   - The remote operating system or server crashed or is unresponsive
   - An admin network issue occurred
2. Address the cause identified in step 1.
   - The remote operating system or server crashed or is unresponsive

     Manually perform switch or disconnection using the mc_ctl command.
   - An admin network issue occurred

     Refer to "Chapter 4 Action Required when an Error Occurs in Database Multiplexing Mode", and recover the database multiplexing system.

3.7 Monitoring Mirroring Controller Messages

The messages that are output by Mirroring Controller are output to both the database server and the arbitration server. If the automatic switch fails, for example, an important message related to the continuation of the operation may be output, so ensure that the system log messages are monitored.

If the arbitration server is used for automatic degradation, monitor messages on both the database server and the arbitration server.

Message output destination on the database server

Messages are output to the system log.

Message output destination on the arbitration server

Linux

Messages are output to the system log.

Windows

Messages are output to the event log.

Point

- To monitor message types considered to be important, an operating system setting must be configured beforehand. Refer to the operating system manuals, check if the message is of a message type that is monitored to be output to the system log, and configure the setting if required.

- If the heartbeat_error_action parameter in *serverIdentifier*.conf is set to "message", only message output is performed when a heartbeat abnormality is detected during heartbeat monitoring of operating systems and servers - automatic switch/disconnection is not performed. Therefore users need to monitor the messages. Refer to "3.6 Action Required when a Heartbeat Abnormality is Detected" for details.

Display format on the database server

```
programName[processId]: messageType:messageText (messageNumber)
```

Specify the program name in the syslog_ident parameter of the serverIdentifier.*conf* file of the database server.

The message types output by Mirroring Controller, their severity, and their corresponding value in the system log are shown in the table below.

Table 3.1 Message type, severity, and corresponding value in the system log

Message type	Severity	Meaning	System log
INFO	Information	Provides information that does not fall under LOG or NOTICE.	INFO
LOG		Provides information recognized as a particularly important event in tracing the operation history. (Example: Automatic switch is complete)	
NOTICE	Notice	Outputs information that takes into account the user instructions within the program in response to an executed or automatically executed process.	NOTICE
WARNING	Warning	Provides a warning, for example it will soon be impossible to maintain the multiplexing state.	WARNING
ERROR	Error	Reports that an error other than FATAL or PANIC has occurred.	ERROR
FATAL		Reports that an abnormality was detected in multiplexed database systems requiring recovery of the system, and also the content and cause of the abnormality.	CRIT
PANIC		Reports that an abnormality was detected in all multiplexed database systems requiring immediate recovery of the system, and also the content and cause of the abnormality.	ALERT

The message severity has the following meanings:

- Information

  Informational status. A message that was reported by the system is displayed. No action is required.

- Notice

  Informational status, but a message that should be noted is displayed. If necessary, take the actions described in the "Action" section of the message.

- Warning

  No error has occurred, but the user is requested to check, and take action. Take the actions described in the "Action" section of the message.

- Error

  An error has occurred. Take the actions described in the "Action" section of the message.

Display format on the arbitration server

Linux

```
programName[processId]: messageType: messageText (messageNumber)
```

Specify the program name in the syslog_ident parameter of the arbitration.conf file of the arbitration server.

Windows

```
eventSourceName[processId]: messageType: messageText (messageNumber)
```

Specify the event source name in the event_source parameter of the arbitration.conf file of the arbitration server.

The message types output by Mirroring Controller, their severity, and their corresponding value in the output destination log are shown in the table below.

Table 3.2 Message type, severity, and corresponding value in the output destination log

Message type	Severity	Meaning	System log (Linux)	Event log (Windows)
INFO	Information	Provides information not categorized as LOG or NOTICE.	INFO	INFORMATION
LOG		Provides information recognized as a particularly important event in tracing the operation history. (Example: Automatic switch is complete)		
NOTICE	Notice	Outputs information that takes into account the user instructions within the program in response to an executed or automatically executed process.	NOTICE	
WARNING	Warning	Provides a warning, for example it will soon be impossible to perform the arbitration process.	WARNING	WARNING
ERROR	Error	Reports that an error other than FATAL or PANIC has occurred.	ERROR	ERROR
FATAL		Reports that an abnormality was detected in the arbitration server requiring recovery of the system, and also the content and cause of the abnormality.	CRIT	
PANIC		Reports that an abnormality was detected in the arbitration server requiring immediate recovery of the system, and also the content and cause of the abnormality.	ALERT	

The message severity has the following meanings:

- Information

  Informational status. A message that was reported by the system is displayed. No action is required.

- Notice

  Informational status, but a message that should be noted is displayed. If necessary, take the actions described in the "Action" section of the message.

- Warning

  No error has occurred, but the user is requested to check, and take action. Take the actions described in the "Action" section of the message.

- Error

  An error has occurred. Take the actions described in the "Action" section of the message.

3.8 Server Maintenance

To perform maintenance tasks such as periodic server inspections and the application of updates for software products including the operating system, you must perform a planned stop of the server, and then perform the maintenance.

3.8.1 Rolling Updates

In database multiplexing mode, rolling updates, that perform the maintenance for the servers that comprise the cluster system, can be performed while jobs continue.

First, perform the maintenance for the standby server, and then switch the standby server to the primary server. Then, perform the maintenance for the original primary server that was switched to the standby server. This enables maintenance to be performed while jobs continue.

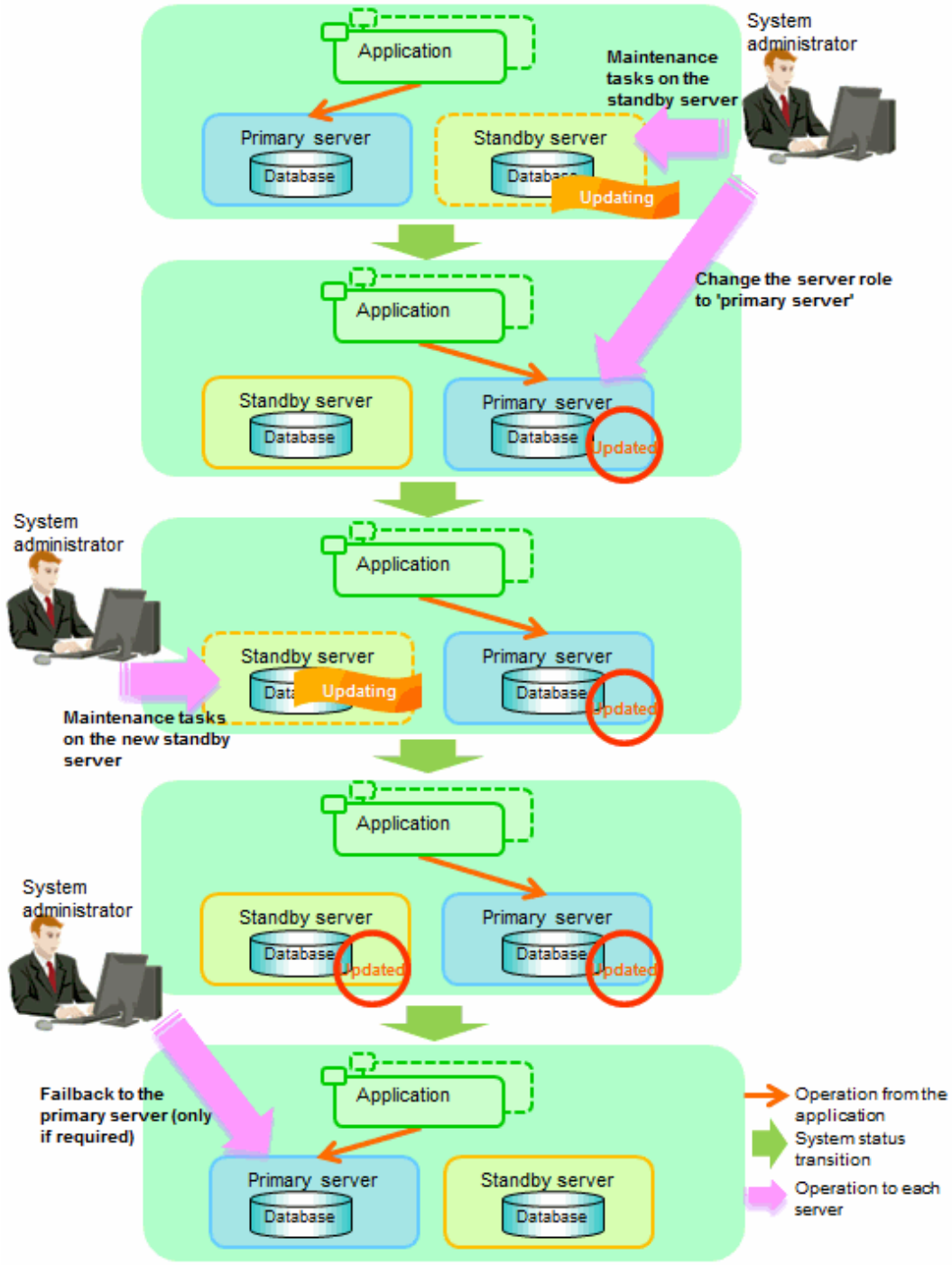
Note that arbitration server maintenance can be performed without affecting database server operation, so it is not necessary to consider rolling update.

See

If the downtime due to the maintenance of the standby server is expected to be long, refer to "Standby server downtime" in "3.9.1 Changes Required when the Standby Server is Stopped".

The flow of a rolling update is shown below.

Figure 3.1 Performing a Rolling Update



Perform the following procedure as shown in the above figure:

Standby server maintenance tasks

1. To perform the maintenance on the standby server, stop Mirroring Controller.

   Example)

   ```
   $ mc_ctl stop -M /mcdir/inst1
   ```

2. Ensure that Mirroring Controller has completely stopped.

   If the multiplexed instances and Mirroring Controller have been configured on the standby server to start and stop automatically when the operating system of the database server is started or stopped, cancel the setting to start and stop automatically.

   Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for information on how to configure the multiplexed instances and Mirroring Controller to start and stop automatically when the operating system of the database server start and stops.

   As the OS superuser, execute the systemctl command to disable automatic start and stop.

   The example below disables automatic start and stop of "mcoi_inst1.service".

   ```
   Example)
   ```

   ```
   # systemctl disable mcoi_inst1.service
   ```

3. Perform maintenance tasks.

4. Create a copy of the primary server instance on the standby server.

   Execute the pg_basebackup command to create data in the standby server by synchronizing with the primary server.

   ```
   Example)
   ```

   ```
   $ pg_basebackup -D /database/inst1 -X fetch --waldir=/transaction/inst1 --progress --verbose -R
   --dbname='application_name=standbyServerName' -h primaryServerHostName -p
   primaryServerPortNumber
   ```

   The procedure for copying the primary server instance to the standby server is the same as the procedure for setting up the standby server.

   Refer to "2.5.2 Creating, Setting, and Registering the Standby Server Instance", and then perform the recovery.

5. Check the settings for automatic start and stop of the multiplexed instances and Mirroring Controller.

   If the multiplexed instances and Mirroring Controller were configured in step 2 to not start and stop automatically when the operating system of the database server starts and stops, then change the settings back. This step can be skipped if automatic start and stop are not required.

   As the OS superuser, execute the systemctl command to enable automatic start and stop.

   The example below disables automatic start and stop of "mcoi_inst1.service".

   ```
   Example)
   ```

   ```
   # systemctl enable mcoi_inst1.service
   ```

6. Start (rebuild) Mirroring Controller on the standby server.

   This operation is required when determining the maintenance tasks on the standby server.

   Enabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode.

   ```
   Example)
   ```

   ```
   $ mc_ctl start -M /mcdir/inst1
   ```

Disabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

Example)

```
$ mc_ctl start -M /mcdir/inst1 -F
```

Point

After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

Switching to the primary server

To perform the maintenance on the primary server, execute the mc_ctl command in the switch mode on the primary server or the standby server.

Example)

```
$ mc_ctl switch -M /mcdir/inst1
```

When the switch is complete, the synchronous_standby_names parameter in the postgresql.conf file of the new primary server will be commented as follows:

Example)

```
#synchronous_standby_names = 'primary'
```

New standby server maintenance tasks

1. Stop the Mirroring Controller.

   On the new standby server (the primary server before the switch), execute the mc_ctl command in stop mode.

   If automatic start and stop of Mirroring Controller has been configured using systemd, do not use the mc_ctl command, but instead use the systemctl command. Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for details.

   Example)

   ```
   $ mc_ctl stop -M /mcdir/inst1
   ```

2. Ensure that Mirroring Controller has completely stopped.

   If the multiplexed instances and Mirroring Controller have been configured on the new standby server to start and stop automatically when the operating system of the database server is started or stopped, cancel the setting to start and stop automatically now.

   See

   Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for information on how to configure the multiplexed instances and Mirroring Controller to start and stop automatically when the operating system of the database server starts and stops.

   As the OS superuser, execute the systemctl command to disable automatic start and stop.

   The example below disables automatic start and stop of "mcoi_inst1.service".

   Example)

   ```
   # systemctl disable mcoi_inst1.service
   ```

3. Perform the maintenance on the new standby server that was stopped.

4. Create a copy of the new primary server instance on the new standby server.

   Execute the pg_basebackup command to create data in the new standby server by synchronizing with the new primary server.

   Example)

```
$ pg_basebackup -D /database/inst1 -X fetch --waldir=/transaction/inst1 --progress --verbose -R
--dbname='application_name=standbyServerName' -h primaryServerHostName -p
primaryServerPortNumber
```

See

The procedure for copying the primary server instance to the standby server is the same as the procedure for setting up the standby server.

Refer to "2.5.2 Creating, Setting, and Registering the Standby Server Instance", and then perform the recovery.

5. Check the settings for automatic start and stop of the multiplexed instances and Mirroring Controller.

   If the multiplexed instances and Mirroring Controller were configured in step 2 to not start and stop automatically when the operating system of the database server starts and stops, then change the settings back. This step can be skipped if automatic start and stop are not required.

   As the OS superuser, execute the systemctl command to enable automatic start and stop.

   The example below disables automatic start and stop of "mcoi_inst1.service".

   Example)

```
# systemctl enable mcoi_inst1.service
```

6. After the maintenance is complete, edit the following parameters in the postgresql.conf file of the standby server as required.

   Copying an instance results in the value of the synchronous_standby_names parameter becoming the specified value on the primary server. Therefore, correct it to the specified value on the standby server. If the parameter was commented out, then you must uncomment it.

7. On the standby server, start (rebuild) Mirroring Controller.

   Enabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode.

   Example)

```
$ mc_ctl start -M /mcdir/inst1
```

   Disabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

   Example)

```
$ mc_ctl start -M /mcdir/inst1 -F
```

Point

After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

Failback of the Primary Server

Revert the primary server and standby server to the original server configuration. Do this to execute the main job on the previous primary server. Refer to "4.1.1.3 Failback of the Primary Server" for details.

Note

Obtain a backup as soon as this task is complete.

3.8.2 Stopping for Maintenance

Perform this procedure to stop all servers for periodic inspections, for example. On the server on which Mirroring Controller is running, execute the mc_ctl command in stop mode to stop the instance and Mirroring Controller.

If automatic start and stop of Mirroring Controller has been configured using systemd, do not use the mc_ctl command, but instead use the systemctl command. Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for details.

After that, on the server where the Mirroring Controller arbitration process is running, execute the mc_arb command in stop mode to stop the Mirroring Controller arbitration process.

Stopping Mirroring Controller

```
Example)
```

```
$ mc_ctl stop -M /mcdir/inst1 -a
```

Stopping the Mirroring Controller arbitration process

Linux

```
Example)
```

```
$ mc_arb stop -M /mcarb_dir/arbiter1
```

Windows

```
Example)
```

```
> mc_arb stop -M D:\mcarb_dir\arbiter1
```

3.8.3 Arbitration Server Maintenance

Arbitration server maintenance can be performed without affecting database server operation.

Follow the procedure below to perform arbitration server maintenance.

1. Execute the mc_arb command in stop mode to forcibly stop the Mirroring Controller arbitration process.

   Linux

   ```
   Example)
   ```

   ```
   $ mc_arb stop -M /mcarb_dir/arbiter1 -e
   ```

   Windows

   ```
   Example)
   ```

   ```
   > mc_arb stop -M D:\mcarb_dir\arbiter1 -e
   ```

2. Perform maintenance tasks.
3. Execute the mc_arb command in start mode to restart the Mirroring Controller arbitration process.

Linux

Example)

```
$ mc_arb start -M /mcarb_dir/arbiter1
```

Windows

Example)

```
> mc_arb start -M D:\mcarb_dir\arbiter1
```

4. Execute the mc_arb command in status mode to check that the arbitration server is connected to the database server.

   The example below executes the mc_arb command, and shows the status.

   Linux

   Example)

```
$ mc_arb status -M /mcarb_dir/arbiter1

server_id    host            status
--------------------------------------
server1      192.0.3.100     online
server2      192.0.3.110     online
```

   Windows

   Example)

```
> mc_arb status -M D:\mcarb_dir\arbiter1

server_id    host            status
--------------------------------------
server1      192.0.3.100     online
server2      192.0.3.110     online
```

5. Check the command output.

   Items to be checked

   Check that the output status is "online" on both lines.

3.9 Changes in Operation

The following changes in operation may be required:

- Changes required when the standby server is stopped
- Changing from single server mode to database multiplexing mode
- Changing from database multiplexing mode to single server mode
- Changing to database multiplexing mode when the arbitration server is used for automatic degradation
- Changing parameters
- Uninstalling in the database multiplexing mode

3.9.1 Changes Required when the Standby Server is Stopped

Operation when the standby server is stopped

Before performing maintenance for the primary server instance when the standby server has been stopped, stop Mirroring Controller on the primary server, comment out the synchronous_standby_names parameter in the postgresql.conf file of the primary server, and then execute the pg_ctl command in reload mode.

If this operation is not performed, operations performed on the primary server for the instance will remain in a wait state.

See

Refer to "pg_ctl" in Reference for information on the command.

Standby server downtime

If you specified the synchronous_standby_names parameter of the postgresql.conf file and then the standby server instance is stopped, consider the points below.

- The wal_sender_timeout parameter in the postgresql.conf file

  If the standby server is stopped after the timeout set in this parameter was exceeded, an error stating that the transaction log could not be received may be output to the primary server system log, and all transaction logs that should be transferred to the standby server are accumulated.

- The wal_keep_size parameter in the postgresql.conf file

  If a transaction log that exceeds the value set in this parameter was generated while the standby server was stopped, the transaction log may be deleted.

  Additionally, setting this parameter requires consideration regarding stabilization of the database multiplexing mode. Refer to "2.11.1 Tuning to Stabilize the Database Multiplexing Mode" for details.

Note

The standby server must be rebuilt if the pending transaction log to be transferred to the standby server is lost when the standby server is started after the maintenance task is complete.

Take the action advised in the recovery operation that starts from "4.1.1.1.3 Identify cause of error and perform recovery" through to "4.1.1.2 Rebuild the Standby Server".

3.9.2 Changing from Single Server Mode to Database Multiplexing Mode

The procedure for switching single server mode to database multiplexing mode for the purposes of high reliability and load distribution of the system is explained below.

This procedure is equivalent to the setup procedure explained in "Chapter 2 Setting Up Database Multiplexing Mode".

Note

If the data storage destination directory name is not comprised of ASCII characters

Stop the application job and then migrate to a directory with a name that uses only ASCII characters:

1. Stop the database instance on the primary server.
2. Change the name of the data storage destination directory to one that uses only ASCII characters.

See

When encrypting the storage data, refer to "Database Multiplexing Mode" in the Operation Guide, and then perform the setup for encryption on the primary and standby servers.

1. Install on the arbitration server

   Perform this step only if the arbitration server is used for automatic degradation.

   Install the Server Assistant on the server where the Mirroring Controller arbitration process is started.

   Refer to "Installation" in the Installation and Setup Guide for Server Assistant for information on how to install the Server Assistant.

2. Install on the standby server

   Install Fujitsu Enterprise Postgres on the server to be started as the standby server.

   Refer to "Installation" in the Installation and Setup Guide for Server for information on how to install Fujitsu Enterprise Postgres.

   Use ASCII characters in the data storage destination directory.

3. Stop the application jobs

   Stop the application jobs to be connected to the primary server.

4. Change the primary server settings

   To allow connections from the server to be started as the standby server, configure the settings in step 2 and thereafter of "2.4.2 Creating, Setting, and Registering the Primary Server Instance" on the primary server.

5. Set up the arbitration server

   Refer to "2.3 Setting Up the Arbitration Server" for details.

   Perform this step only if the arbitration server is used for automatic degradation.

6. Set up database multiplexing mode on the primary server

   Refer to "2.4.1 Setting Up Database Multiplexing Mode on the Primary Server" for details.

7. Set up database multiplexing mode on the standby server

   Refer to "2.5.1 Setting Up Database Multiplexing Mode on the Standby Server" for details.

8. Create the standby server instance and start it

   Refer to "2.5.2 Creating, Setting, and Registering the Standby Server Instance" for details.

After the above steps are completed, refer to the remaining explanations in "Chapter 2 Setting Up Database Multiplexing Mode" and ensure that the required settings and operations are completed.

3.9.3 Changing from Database Multiplexing Mode to Single Server Mode

The procedure for stopping database multiplexing mode and changing to single server mode is explained below.

Some tasks must be performed on the database server, and others must be performed on the arbitration server.

The tasks on the arbitration server are required only if the arbitration server is used for automatic degradation.

Tasks on the database server

1. Determine the server for which the instance is to be stopped, and switch this server

   Determine the server that is to be excluded as the database multiplexing mode target, and for which the instance is to be stopped.

   If the server for which the instance is to be stopped is the primary server, execute the mc_ctl command in the switch mode to switch the standby server to the primary server.

   The standby server after the switch is complete will be the server for which the instance is to be stopped.

   If the server for which the instance is to be stopped is the standby server, there is no need to perform the switch operation.

   Example)

   ```
   $ mc_ctl switch -M /mcdir/inst1
   ```

2. Stop Mirroring Controller and the instance, and delete the file resources

   On the server that was determined in step 1, execute the mc_ctl command in stop mode to stop Mirroring Controller and the instance.

If automatic start and stop of Mirroring Controller has been configured using systemd, do not use the mc_ctl command, but instead use the systemctl command. Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for details.

Example)

```
$ mc_ctl stop -M /mcdir/inst1
```

Then, delete the following file resources:

- Data storage destination directory
- Mirroring Controller management directory

Example)

```
$ rm -rf /database/inst1
$ rm -rf /mcdir/inst1
```

See

Refer to "Security-Related Notes" in the Operation Guide for details on deleting the data securely.

3. Stop the application jobs

   Stop the application jobs to be connected to the primary server.

4. Stop Mirroring Controller and the instance on the primary server

   Execute the mc_ctl command in stop mode on the primary server.

   If automatic start and stop of Mirroring Controller has been configured using systemd, do not use the mc_ctl command, but instead use the systemctl command. Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for details.

   Example)

   ```
   $ mc_ctl stop -M /mcdir/inst1
   ```

5. Delete the database multiplexing mode settings that were configured for the primary server instance.

   Reset the postgresql.conf file parameters to their values before the database multiplexing operation was set.

   Delete the file resources from the Mirroring Controller management directory.

   If the backup operation was performed, delete the following resources:

   - Mirroring Controller management directory backup data obtained in database multiplexing mode
   - Instance backup data obtained in database multiplexing mode

   Additionally, if the primary_conninfo parameter is set in the postgresql.auto.conf file, execute the ALTER SYSTEM RESET statement to delete the setting.

   Example)
   An example execution of the psql command is shown below.

   ```
   postgres=# ALTER SYSTEM RESET primary_conninfo;
   ```

   After these actions are performed, ensure that the backup data is collected when starting the single operation.

See

- Refer to "Security-Related Notes" in the Operation Guide for details on deleting the data securely.
- Refer to "2.14 Backup Operation" for details on the backup operation.

- Refer to "Appendix A Parameters" for details on the postgresql.conf file parameters.

Point

In the above procedure, if the postgresql.conf file of the single primary server can be changed by reloading the file, the operation mode can be changed without stopping the application job.

In that case, execute the mc_ctl command in stop mode with the --mc-only option specified to stop only Mirroring Controller in relation to stopping the primary server.

Tasks on the arbitration server

Linux

1. Execute the mc_arb command in stop mode to stop the Mirroring Controller arbitration process.

   Example)

   ```
   $ mc_arb stop -M /mcarb_dir/arbiter1
   ```

2. Delete the Mirroring Controller arbitration process management directory.

   Example)

   ```
   $ rm -rf /mcarb_dir/arbiter1
   ```

Windows

1. Execute the mc_arb command in stop mode to stop the Mirroring Controller arbitration process.

   Example)

   ```
   > mc_arb stop -M D:\mcarb_dir\arbiter1
   ```

2. Unregister the Mirroring Controller arbitration process from the Windows service.

   Execute the mc_arb command in unregister mode to unregister the Mirroring Controller arbitration process from the Windows service.

   Example)

   ```
   > mc_arb unregister -M D:\mcarb_dir\arbiter1
   ```

3. Delete registrations related to the event log

   If error logs are output to the event log in "2.2.2.1 Preparing to Output Error Logs to the Event Log (Windows)", delete the registered event source name for each instance.

   Example)

   ```
   > regsvr32 /u /i:"Mirroring Controller arbtier1" "c:\Program Files\Fujitsu
   \fsepv<x>assistant64\lib\mcarbevent.dll"
   ```

   Note that "<*x*>" indicates the product version.

4. Delete the Mirroring Controller arbitration process management directory.

   Example)

   ```
   > rmdir /S /Q D:\mcarb_dir\arbiter1
   ```

3.9.4 Changing to Database Multiplexing Mode when the Arbitration Server is Used for Automatic Degradation

This section provides the procedure to change to database multiplexing mode using the Mirroring Controller only on the database server when the arbitration server is used for automatic degradation.

Some tasks must be performed on the database server, and others must be performed on the arbitration server.

Tasks on the arbitration server

1. Set up the arbitration server.

   Refer to "2.3 Setting Up the Arbitration Server" for information on how to set up the arbitration server.

Tasks on the database server

1. On the server where Mirroring Controller is running, execute the mc_ctl command in stop mode to stop Mirroring Controller on the primary server and standby server.

   Example)

   ```
   $ mc_ctl stop -M /mcdir/inst1 -a --mc-only
   ```

2. Edit the network.conf file of the primary server and standby server to add the information of the arbitration server.

   Refer to "A.3 Network Configuration File" for details.

   The definition example of the network.conf file of the primary server is shown below:

   Example)

   The IDs of the primary server and standby server are set to "server1" and "server2", and their port numbers are set to "27540" and "27541". The ID of the server of the Mirroring Controller arbitration process is set to "arbiter", and its port number is set to "27541".

   ```
   server1 192.0.2.100,192.0.3.100 27540,27541 server
   server2 192.0.2.110,192.0.3.110 27540,27541 server
   arbiter 192.0.3.120 27541 arbiter
   ```

   Note

   - Ensure that the port numbers set for the primary server, standby server, and arbitration server do not conflict with other software. Also do not configure the same segment for the admin network and arbitration network.
   - If the server type is "server", two IP addresses or host names, and two port numbers need to be specified in the following order:
     - IP address or host name of the database server used as the admin network
     - IP address or host name of the database server used as the arbitration network
     - Port number of the database server used as the admin network
     - Port number of the database server used as the arbitration network
   - If the server type is "arbiter", specify the IP address or host name set for the my_address parameter and the port number set for the port parameter in arbitration.conf.

3. Edit the *serverIdentifier*.conf file of the primary server and standby server to add parameters required for the operation where the arbitration server is used for automatic degradation.

   Refer to "A.4.1 Server Configuration File for the Database Servers" for information on the parameters required when the arbitration server is used for automatic degradation.

4. On the primary server and standby server, execute the mc_ctl command in start mode to start the Mirroring Controller process.

   Example)

```
$ mc_ctl start -M /mcdir/inst1 --mc-only
```

Common tasks

1. Check the connection status from the database server or arbitration server.

   Refer to "2.8 Checking the Connection Status" for details.

3.9.5 Changing Parameters

Stop Mirroring Controller before editing the Mirroring Controller server configuration file and network configuration file.

If the Mirroring Controller process crashes or becomes unresponsive, restart is performed automatically by the Mirroring Controller monitoring process, and the configuration file is reloaded. Therefore, if the configuration file was being edited, unintended behavior will occur.

3.9.6 Uninstalling in Database Multiplexing Mode

This section explains how to uninstall Fujitsu Enterprise Postgres on a server using database multiplexing mode.

Some tasks must be performed on the database server, and others must be performed on the arbitration server.

The tasks on the arbitration server are required only if the arbitration server is used for automatic degradation.

Tasks on the database server

1. Stop the multiplexed instances and Mirroring Controller

   Refer to "3.2 Starting and Stopping Mirroring Controller" for information on how to stop the instance.

2. Uninstall Fujitsu Enterprise Postgres

   Refer to "Uninstallation" in the Installation and Setup Guide for Server for information on how to uninstall Fujitsu Enterprise Postgres.

Tasks on the arbitration server

Refer to "Uninstallation" in the Installation and Setup Guide for Server Assistant, and uninstall the Server Assistant.

Chapter 4 Action Required when an Error Occurs in Database Multiplexing Mode

This chapter describes the action required if an error occurs in database multiplexing mode.

In database multiplexing mode, when an error is detected, the switch or disconnection of the standby server is performed automatically, so that only the primary server starts degrading. In this case, the recovery tasks will be required for the standby server on which the switch or disconnection was performed.

Other possible cases are as follows:

- When automatic switch fails
- When automatic disconnection fails
- When all servers or instances were stopped

4.1 Action Required when Server Degradation Occurs

If the server has started degrading, the recovery tasks will vary depending on whether the cause was the switch (failover or switchover), or the disconnection.

Execute the mc_ctl command in status mode, or refer to the system log, and check if the cause of the server degradation was the switch or the disconnection.

In the example below, the mc_ctl command is executed in status mode.

If a switch has occurred, "switched" (the switch is complete and the server is in a degrading state) is displayed for "mirroring status".

Example)

```
$ mc_ctl status -M /mcdir/inst1
mirroring status
----------------
switched

   :
```

If a disconnection has occurred, "not-switchable" (disconnection was performed so the server cannot be switched) is displayed for "mirroring status".

Example)

```
$ mc_ctl status -M /mcdir/inst1
mirroring status
----------------
not-switchable

  :
```

Note

If Mirroring Controller detects any errors on the server on which operations are continuing during recovery to database multiplexing mode from a degrading operation state, perform the procedure in "4.1.3 Addressing Errors During Degrading Operation", and then recover to database multiplexing mode.

4.1.1 Operations when the Server has Started Degrading after a Switch has Occurred

This section explains the operations when the server has started degrading after a switch has occurred.

Note

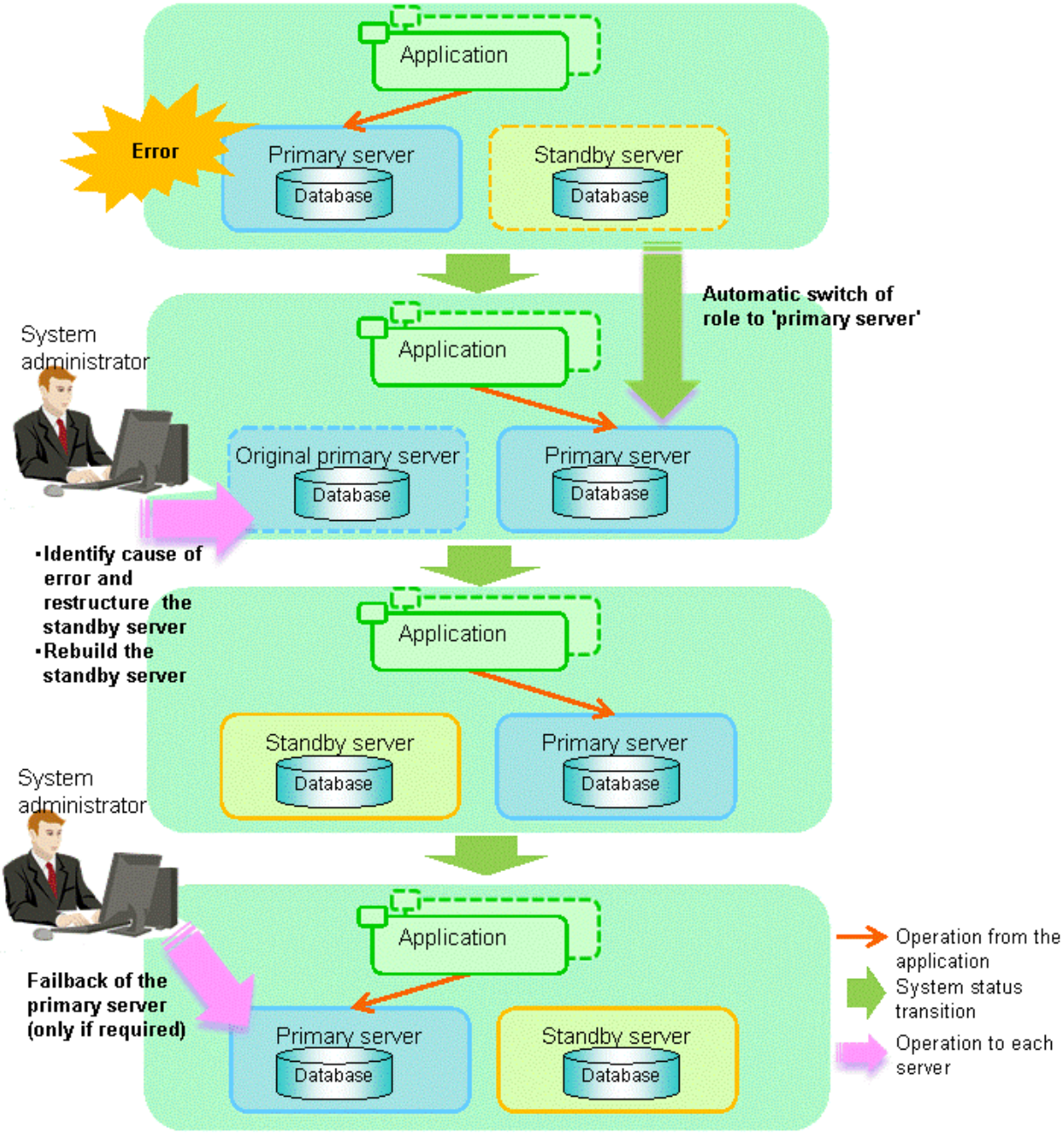
- After a switch has occurred as a result of an abnormality on the primary server, the database will not have a multiplexed configuration until the standby server is rebuilt. Remove the cause of the error as quickly as possible, and then rebuild the standby server.
- If the reference job was executed on the standby server, and the servers are switched because an error occurred on the primary server, the load is concentrated on the new primary server. Accordingly, pause the reference job on the original standby server, rebuild the original primary server as the new standby server, and then resume the reference job for the new standby server.
- If the instance on the new primary server is stopped before the original primary server where the error occurred is rebuilt as the new standby server, a split brain occurs at startup from the instance on the original primary server. Therefore, start the instance on the new primary server before rebuilding the standby server.

If the switch occurred and the server has started degrading, perform the following operations to recover the standby server and revert it to its original state:

- Identify Cause of Error and Restore the Standby Server
- Rebuild the Standby Server
- Failback of the Primary Server (only if required)

The flow of these operations is shown in the figure below.

Figure 4.1 Flow of operations



4.1.1.1 Identify Cause of Error and Restore the Standby Server

Perform the recovery according to the following procedure:

1. Stop Mirroring Controller
2. Recovery of the Mirroring Controller management directory
3. Identify cause of error and perform recovery

4.1.1.1.1 Stop Mirroring Controller

Execute the mc_ctl command in stop mode for the original primary server on which the error occurred.

If automatic start and stop of Mirroring Controller has been configured using systemd, do not use the mc_ctl command, but instead use the systemctl command. Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for details.

Example)

```
$ mc_ctl stop -M /mcdir/inst1
```

This also stops the instance that is required to perform the recovery.

Note

If the instance does not stop, refer to "Actions in Response to Failure to Stop an Instance" in the Operation Guide, and then stop the instance.

Then, specify the -e option in the above command to forcibly stop Mirroring Controller.

4.1.1.1.2 Recovery of the Mirroring Controller management directory

Copy the files in the Mirroring Controller management directory from the backup data, and then perform the recovery.

4.1.1.1.3 Identify cause of error and perform recovery

Refer to the system log of the primary server and the standby server to identify the cause of the error, and then perform recovery.

The following commands can be used to recover a standby server. Select depending on the recovery and the situation.

- pg_basebackup

  Creates a copy of all resources of the primary server instance.

- pg_rewind

  Creates a copy of only the updated files on the new primary server. For this reason, if this command is used to incorporate a new standby server, recovery time can be shortened. To use this command to build the original primary server as a new standby server, at least one of the following must be met:

  a. Checksums were enabled when an instance was created, or

  b. The wal_log_hints parameter of postgresql.conf was enabled when an instance was started.

  Additionally, full_page_writes must be enabled, which is its default value.

See

- Refer to "pg_basebackup" in "Reference" in the PostgreSQL Documentation for information on the pg_basebackup command.
- Refer to "pg_rewind" in "Reference" in the PostgreSQL Documentation for information on the pg_rewind command.

The example below executes the pg_rewind command to perform recovery by synchronizing data on the original primary server with the new primary server.

1. Wait for the application of unapplied update transaction logs on the new primary server.

   Execute the SQL below on the new primary server, and wait until the result is false.

   ```
   # select pg_is_in_recovery();
   ```

   Example)

   ```
   $ psql -h hostNameOfNewPrimaryServer -p portNumOfNewPrimaryServer -d dbName -c "select
   pg_is_in_recovery();"
   ```

   Any database can be connected to.

Note

If the pg_rewind command is executed immediately after promotion of the new primary server, the processing in steps 1 and 2 is required. If update-type SQL can be executed on the new primary server and checkpoint processing is executed after promotion, the processing in steps 1 and 2 will not be necessary.

2. Update the timeline ID.

   Execute checkpoint processing, and update the timeline ID.

```
$ psql -h hostNameOfNewPrimaryServer -p portNumOfNewPrimaryServer -d dbName -c "checkpoint;"
```

   Any database can be connected to.

3. Create a copy of the new primary server instance in the original primary server (new standby server).

   Execute the pg_rewind command to synchronize the new standby server data with the new primary server.

   Example)

```
$ pg_rewind -D /database/inst1 -R --source-server='user=userName host=newPrimaryServerHostName
port=newPrimaryServerPortNumber dbname=dbName application_name=newStandbyServerName'
```

Note

- Use the pg_rewind command with the -R option to create a standby.signal file. If you do not create the standby.signal file, the Mirroring Controller cannot be started as a standby server.
- If using a method that requires password authentication for connections to the primary server, you will need to ensure that authentication is performed automatically. If the -R option is specified for the pg_rewind command and the password parameter is specified for the --dbname option, the pg_rewind command will set the password in the primary_conninfo parameter in postgresql.auto.conf file, enabling connections to be performed automatically.

  If a password is not set in the primary_conninfo parameter in postgresql.auto.conf file, it will be necessary to create a .pgpass file in the home directory of the instance administrator user, and specify a password for the replication database.
- If you need to set a connection string other than host, port and application_name, include it in the setting of the primary_conninfo parameter.
- The primary_conninfo parameter should not be set in the postgresql.conf file, but only in the postgresql.auto.conf file using the pg_rewind command.

4. Specify parameters in the postgresql.conf file of the original primary server (new standby server).

   Set the parameters required for the standby server in postgresql.conf.

   Refer to "Table 2.5 Parameters" for information on the parameters to set in postgresql.conf.

See

- Refer to "Hot Standby" in the PostgreSQL Documentation for details on the standby.signal file.
- Refer to "Setting Up a Standby Server" in the PostgreSQL Documentation for details on the primary_conninfo.

Note

A new timeline is branched for the new primary server due to promotion, so 'latest' needs to be specified for the recovery_target_timeline parameter so that the old primary server (new standby server) follows the new primary server.

4.1.1.2 Rebuild the Standby Server

The starting of the recovered original primary server as the standby server is referred to as the "standby server rebuild".

On the original primary server, start Mirroring Controller and the instance.

Enabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode.

Example)

```
$ mc_ctl start -M /mcdir/inst1
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

Example)

```
$ mc_ctl start -M /mcdir/inst1 -F
```

Point

After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

4.1.1.3 Failback of the Primary Server

To revert the primary server and standby server to the original server configuration after rebuilding the standby server, perform failback for the primary server.

Do this to execute the main job on the previous primary server.

Perform the following procedure:

1. Failback of the primary server

   Execute the mc_ctl command in switch mode on the primary server or the standby server.

   Example)

   ```
   $ mc_ctl switch -M /mcdir/inst1
   ```

   After executing the mc_ctl command in switch mode, the status will be as follows:

   Example)

   ```
   $ mc_ctl status -M /mcdir/inst1
   mirroring status
   ----------------
   switched
   server_id  host_role                  host         host_status  db_proc_status       disk_status
   ------------------------------------------------------------------------------------------------
   server1    primary                  192.0.2.100 normal        abnormal(postmaster) normal
   server2    none(inactivated primary) 192.0.2.110 normal        abnormal(postmaster) normal
   ```

2. Stop the original primary server

   On the original primary server, execute the mc_ctl command in stop mode to stop Mirroring Controller and the instance.

   If automatic start and stop of Mirroring Controller has been configured using systemd, do not use the mc_ctl command, but instead use the systemctl command. Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for details.

   Example)

```
$ mc_ctl stop -M /mcdir/inst1
```

3. Create a copy of the new primary server instance in the original primary server (new standby server)

   Execute the pg_basebackup command to create data in the new standby server by synchronizing with the new primary server.

   Example)

```
$ pg_basebackup -D /database/inst1 -X fetch --waldir=/transaction/inst1 --progress --verbose -R
--dbname='application_name=standbyServerName' -h primaryServerHostName -p
primaryServerPortNumber
```

See

The procedure for copying the new primary server instance to the new standby server is the same as the procedure for setting up the new standby server.

Refer to "2.5.2 Creating, Setting, and Registering the Standby Server Instance", and then perform the recovery.

4. Rebuild the standby server

   On the standby server, start Mirroring Controller and the instance.

   Enabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode.

   Example)

```
$ mc_ctl start -M /mcdir/inst1
```

   Disabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

   Example)

```
$ mc_ctl start -M /mcdir/inst1 -F
```

Point

After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

4.1.2 Operations when the Server has Started Degrading after a Disconnection has Occurred

This section explains the operations when the server has started degrading after a disconnection has occurred.

Note

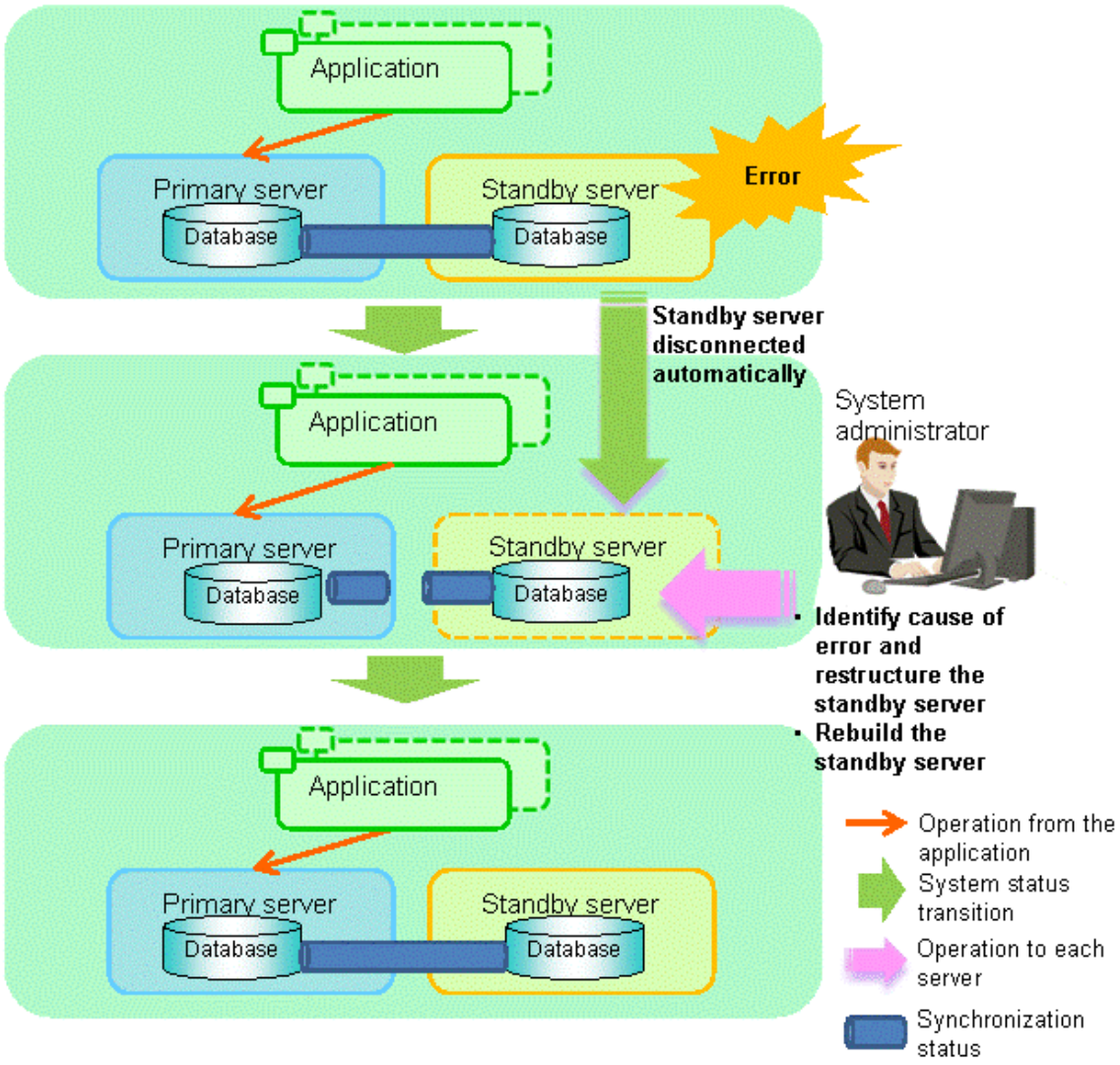
After a disconnection has occurred as a result of an abnormality on the standby server, the database will not have a multiplexed configuration until the standby server is rebuilt. Remove the cause of the error as quickly as possible, and then rebuild the standby server.

If the disconnection occurred and the server has started degrading, perform the following operations to recover the standby server and revert it to its original state:

- Identify Cause of Error and Restore the Standby Server
- Rebuild the Standby Server

The flow of these operations is shown in the figure below.

Figure 4.2 Flow of operations



4.1.2.1 Identify Cause of Error and Restore the Standby Server

Perform the recovery according to the following procedure:

1. Stop Mirroring Controller
2. Recovery of the Mirroring Controller management directory
3. Identify cause of error and perform recovery

4.1.2.1.1 Stop Mirroring Controller

Execute the mc_ctl command in stop mode for the standby server on which the error occurred.

If automatic start and stop of Mirroring Controller has been configured using systemd, do not use the mc_ctl command, but instead use the systemctl command. Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for details.

Example)

```
$ mc_ctl stop -M /mcdir/inst1
```

This also stops the instance that is required to perform the recovery.

Note

If the instance does not stop, refer to "Actions in Response to Failure to Stop an Instance" in the Operation Guide, and then stop the instance.

Then, specify the -e option in the above command to forcibly stop Mirroring Controller.

4.1.2.1.2 Recovery of the Mirroring Controller management directory

Copy the files in the Mirroring Controller management directory from the backup data, and then perform the recovery.

4.1.2.1.3 Identify cause of error and perform recovery

Refer to the system logs of the primary server and the standby server to identify the cause of the error, and then perform recovery.

Execute the pg_basebackup command to perform recovery by synchronizing data in the primary server with the standby server.

```
Example)
```

```
$ pg_basebackup -D /database/inst1 -X fetch --waldir=/transaction/inst1 --progress --verbose -R --
dbname='application_name=standbyServerName' -h primaryServerHostName -p primaryServerPortNumber
```

See

This recovery procedure is the same as the procedure for setting up the standby server.

Refer to "2.5.2 Creating, Setting, and Registering the Standby Server Instance", and then perform the recovery.

4.1.2.2 Rebuild the Standby Server

Start the Mirroring Controller and the instance of the standby server, and rebuild the standby server.

Enabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode.

```
Example)
```

```
$ mc_ctl start -M /mcdir/inst1
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

```
Example)
```

```
$ mc_ctl start -M /mcdir/inst1 -F
```

Point

After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

4.1.3 Addressing Errors During Degrading Operation

This section explains how to address errors that may occur on the server on which operation is continuing during degrading operation triggered by a switch or disconnection.

If needing to recover from backup data

If it is necessary to recover the database using backup data due to data becoming corrupted from disk failure or user operation error, refer to the following for information on recovery to database multiplexing mode:

- Action Required when All Database Servers or Instances Stopped
- Recovering from an Incorrect User Operation

If a temporary error occurs

If a temporary error occurs, such as due to a high load on the server or insufficient system resources, remove the cause of the error and restart Mirroring Controller, and then refer to the following for details on recovery to database multiplexing mode:

- Operations when the Server has Started Degrading after a Switch has Occurred
- Operations when the Server has Started Degrading after a Disconnection has Occurred

See

Refer to "3.2 Starting and Stopping Mirroring Controller" for information on restarting Mirroring Controller.

4.2 Action Required when Automatic Switch Fails

If the system behavior is unstable, for example there are insufficient temporary system resources, the Mirroring Controller automatic switch may fail.

Perform the switch manually using one of the following methods:

- Refer to the procedures in "3.4 Manually Switching the Primary Server".
- In the standby server, execute the mc_ctl command in switch mode with the -force option specified to forcibly perform the switch.

  Example)

```
$ mc_ctl switch -M /mcdir/inst1 --force
```

Point

- Even if connection cannot be established between database servers, it is possible to fence the primary server and forcibly switch by executing the mc_ctl command in switch mode with the --force option specified.
- The primary server is not fenced in the cases below, so stop Mirroring Controller and instances of the primary server database in advance:
  - The --no-fencing option is specified when performing forced switch.
  - The heartbeat_error_action parameter in *serverIdentifier*.conf is set to "message" and the fencing command is not configured to be used (the fencing_command parameter is omitted in *serverIdentifier*.conf).
  - The heartbeat_error_action parameter in *serverIdentifier*.conf is set to "fallback".

See

Recovery to database multiplexing mode

Refer to "4.1.1.2 Rebuild the Standby Server" and "4.1.1.3 Failback of the Primary Server" for information on recovery to database multiplexing mode.

4.3 Action Required when Automatic Disconnection Fails

If the system behavior is unstable, for example there are insufficient system resources such as available memory or free disk space, automatic disconnection using Mirroring Controller may not be possible.

Perform the disconnection manually using one of the following methods:

- Refer to the procedures in "3.5 Manually Disconnecting the Standby Server".
- In the primary server, execute the mc_ctl command in detach mode to perform forced disconnection.

  Example)

```
$ mc_ctl detach -M /mcdir/inst1
```

Point

- Even if connection cannot be established between database servers, it is possible to fence the standby server and forcibly disconnect by executing the mc_ctl command in detach mode.
- In the cases below, stop Mirroring Controller and instances of the standby server database in advance so that the standby server is not fenced:
  - The --no-fencing option is specified when performing forced disconnection.
  - The heartbeat_error_action parameter in *serverIdentifier*.conf is set to "message" and the fencing command is not configured to be used (the fencing_command parameter is omitted in *serverIdentifier.conf*).
  - The heartbeat_error_action parameter in *serverIdentifier*.conf is set to "fallback".

See

Recovery to database multiplexing mode

Refer to "4.1.2.2 Rebuild the Standby Server" for information on recovery to database multiplexing mode.

4.4 Action Required when All Database Servers or Instances Stopped

This section explains what happens when all database servers or instances on the database server have stopped, so jobs cannot continue.

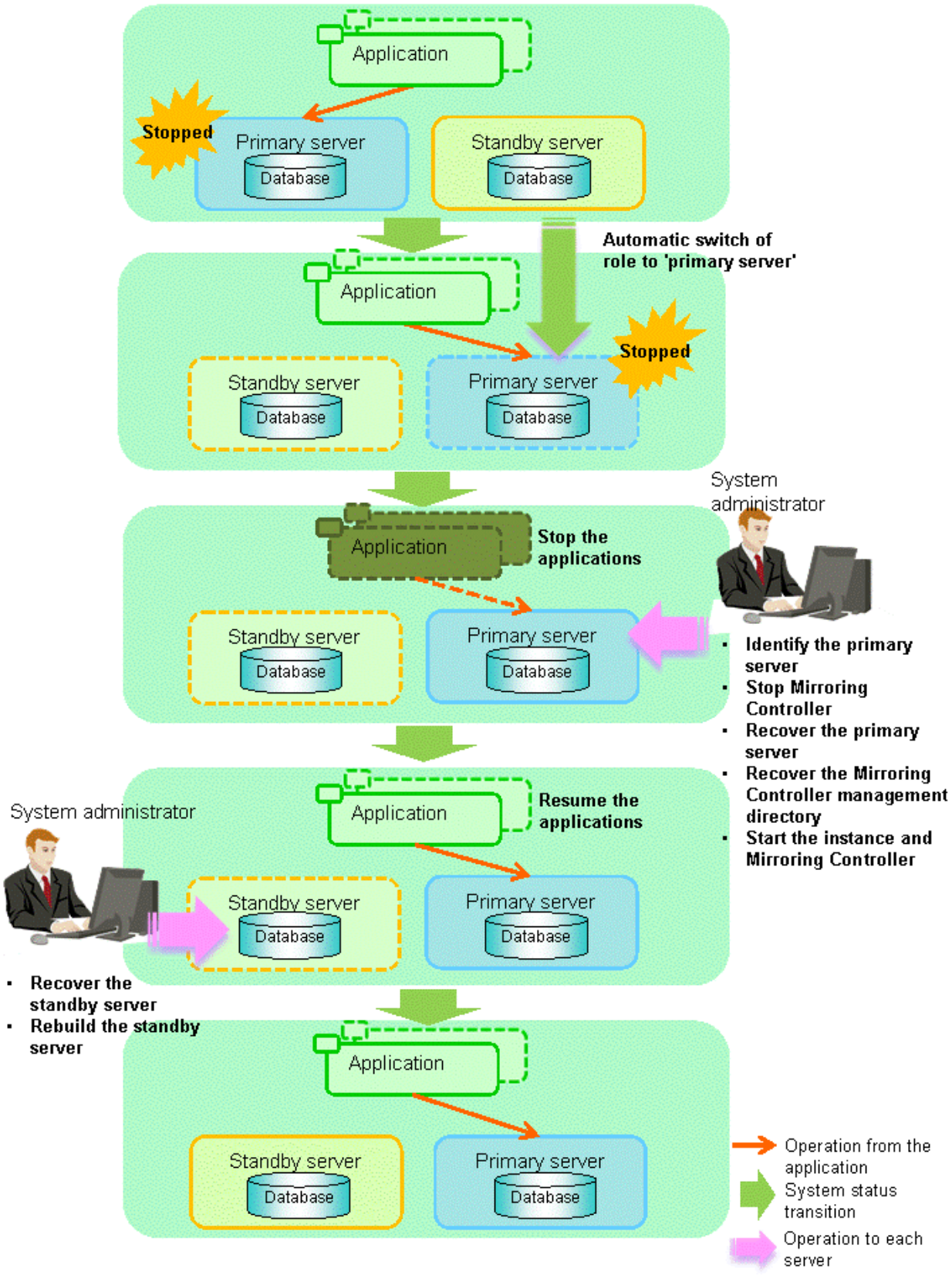
See

Recovery to database multiplexing mode

Refer to "4.1.1.2 Rebuild the Standby Server" and "4.1.1.3 Failback of the Primary Server" for information on recovery to database multiplexing mode.

The flow of these recovery operations is shown in the figure below.

Figure 4.3 Flow of operations



Perform the following procedure:

1. Stop the applications

   Stop the applications from running.

2. Identify the primary server

   Use one of the following methods to identify the primary server that was running before the servers or instances stopped:

   - Refer to the system log on each server and identify the server where the following message was output.

     Message:

     ```
     MirroringControllerOpen[30017]: LOG:  promotion processing completed (MCA00062)
     ```

   - On each server, execute the mc_ctl command in status mode to search the servers for which "none(inactivated primary)" is displayed.

3. Stop Mirroring Controller on the primary server

   Execute the mc_ctl command in stop mode on the primary server.

   If automatic start and stop of Mirroring Controller has been configured using systemd, do not use the mc_ctl command, but instead use the systemctl command. Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for details.

   Example)

   ```
   $ mc_ctl stop -M /mcdir/inst1
   ```

   Note

   Forcibly stopping Mirroring Controller

   If Mirroring Controller does not stop, specify the -e option in the stop mode of the mc_ctl command and then execute the command.

   Example)

   ```
   $ mc_ctl stop -M /mcdir/inst1 -e
   ```

4. Recover the primary server

   First, refer to "Actions when an Error Occurs" in the Operation Guide, and then identify the cause of the error and perform recovery.

   Next, recover the primary server using the recovery method that uses the pgx_rcvall command based on the backup data.

   If the backup operation was performed using the pgx_dmpall command based on the instructions in "2.14.2 Database Backup Operation", perform the following procedure for the recovery:

   a. Perform the following operations on both the primary server and the standby server, and check the server containing the backup data and the archive log that show the latest date.

      - Execute the pgx_rcvall command with the -l option specified and identify the backup data that shows the latest date.
      - Identify the archive log that shows the latest date, as shown below.

        Example)

        ```
        $ ls -ltr backupDataStorageDir/*_wal
        ```

   b. If the latest backup data exists on the standby server, copy (*1) the backup data and overwrite (*2) it to each backup storage destination directory on the primary server.

   c. If the latest archive log and transaction log file exist on the standby server, copy (*1) the archive log and overwrite (*2) it to the backup storage destination directory on the primary server.

   d. Execute the pgx_rcvall command on the primary server, specifying the backup storage destination directory of the primary server.

Note

*1: The backup data may contain a symbolic link, so copy the backup data so that the symbolic link is not converted to an ordinary file (with the tar command, for example).

*2: If you can save a copy of the backup storage destination directory, do so without overwriting it.

See

Refer to "Actions when an Error Occurs" in the Operation Guide for information on the pgx_rcvall command.

5. Recover the Mirroring Controller management directory

   Copy the files in the Mirroring Controller management directory from the backup data, and then perform the recovery.

6. Start the primary server instance and Mirroring Controller

   Enabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode.

   Example)

   ```
   $ mc_ctl start -M /mcdir/inst1
   ```

   Disabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

   Example)

   ```
   $ mc_ctl start -M /mcdir/inst1 -F
   ```

Point

After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

7. Resume applications

   Resume the applications.

8. Stop Mirroring Controller on the standby server

   Execute the mc_ctl command in stop mode on the standby server.

   If automatic start and stop of Mirroring Controller has been configured using systemd, do not use the mc_ctl command, but instead use the systemctl command. Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for details.

   Example)

   ```
   $ mc_ctl stop -M /mcdir/inst1
   ```

9. Recover the standby server

   Refer to "2.5.2 Creating, Setting, and Registering the Standby Server Instance", and then recover (set up) the standby server from the primary server.

10. Rebuild the standby server

    On the standby server, start Mirroring Controller and the instance.

Enabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode.

Example)

```
$ mc_ctl start -M /mcdir/inst1
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

Example)

```
$ mc_ctl start -M /mcdir/inst1 -F
```

Point

After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

4.5 Recovering from an Incorrect User Operation

This section describes how to recover an instance when data has been corrupted due to incorrect user operation.

For example, when data has been corrupted due to incorrect user operation, such as data being unintentionally changed or deleted by an application or command, it is necessary to restore the original data on the primary server and resynchronize with the standby server.

Use the following procedure to perform recovery.

1. Identify the primary server

   Execute the mc_ctl command in status mode on each server, and search for a server for which "primary" or "none(inactivated primary)" is displayed.

2. Stop the applications and commands that caused the incorrect operation to occur

   Stop applications and commands that are running on the primary server. This will minimize the impact caused by the incorrect data.

   Also, if any applications used for reference by the standby server are running, stop them too.

3. Stop the instance and Mirroring Controller

   Stop the instance and Mirroring Controller on both the primary server and standby server.

   Example)

   ```
   $ mc_ctl stop -a -M /mcdir/inst1
   ```

4. Recover the database on the primary server

   Recover the database using the recovery method in which the pgx_rcvall command uses the backup data to recover the database to a restore point prior to the time when the incorrect operation was performed.

   See

   Refer to "Recovering from an Incorrect User Operation" in the Operation Guide for information on using the pgx_rcvall command to recover the database to a restore point, and then perform only the database recovery procedure while the instance is in a stop state.

5. Start the instance and Mirroring Controller

   Start the instance and Mirroring Controller on the primary server.

   Enabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode.

Example)

```
$ mc_ctl start -M /mcdir/inst1
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

Example)

```
$ mc_ctl start -M /mcdir/inst1 -F
```

Point

After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

6. Build the new standby server

   Refer to "2.5 Setting Up the Standby Server" for information on building (setting up) a standby server from the primary server.

Chapter 5 Managing Mirroring Controller Using WebAdmin

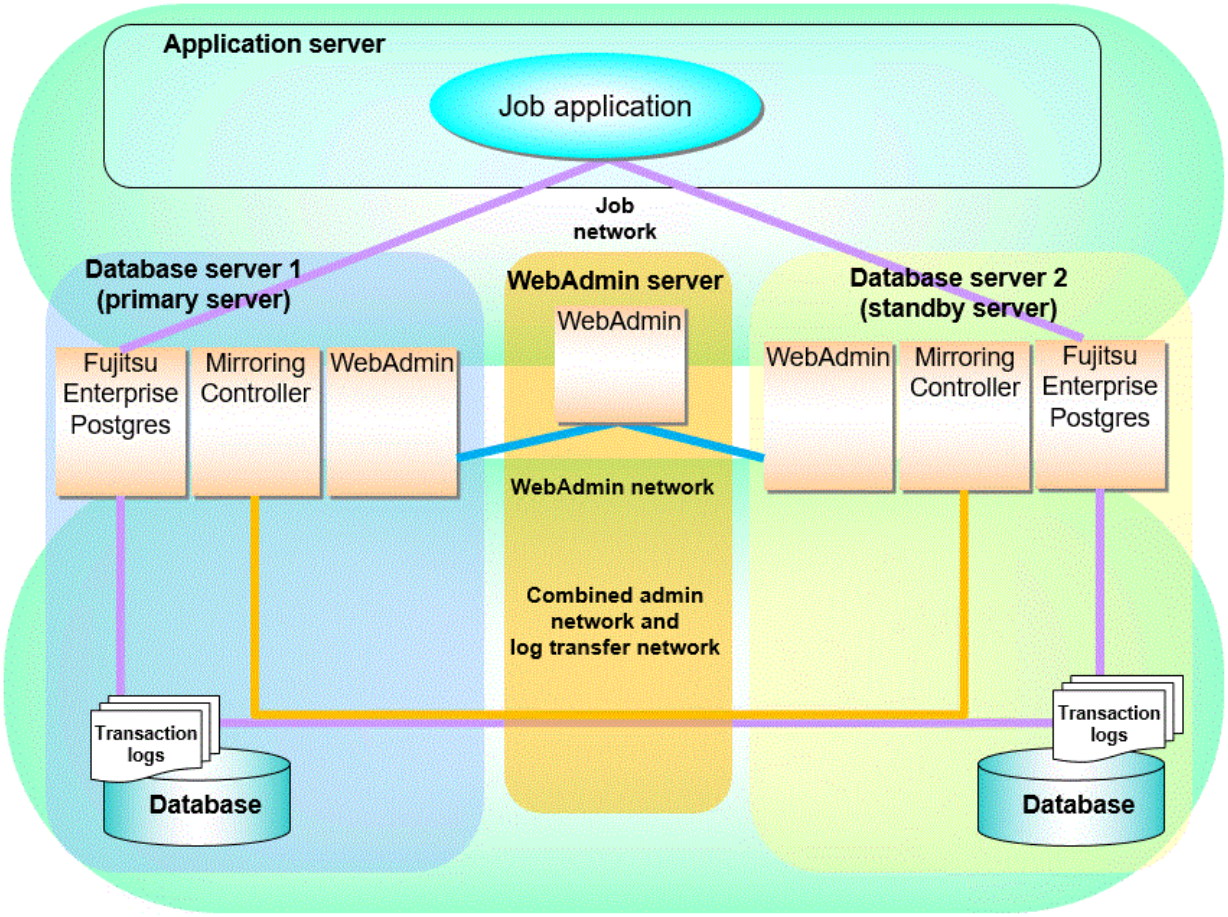
This chapter describes how to set up and manage Mirroring Controller in a streaming replication cluster using WebAdmin.

Mirroring Controller can be used to monitor a streaming replication cluster and perform automatic switching or disconnect synchronous replication when there is an error.

WebAdmin can be used to set up Mirroring Controller in an existing replication cluster. Mirroring Controller can be set up for either synchronous standby instances or asynchronous standby instances.

The configuration of the database multiplexing system built using WebAdmin is shown below:

Figure 5.1 Configuration of database multiplexing operation system using WebAdmin



Point

- If Mirroring Controller is set up to the replication cluster using WebAdmin, the network with the host name (or IP address) specified in [Host name] will be used as the admin network and the log transfer network.
- To use a network other than the job network as the log transfer network, before building the replication cluster specify a host name other than the job network one in [Host name].

Note

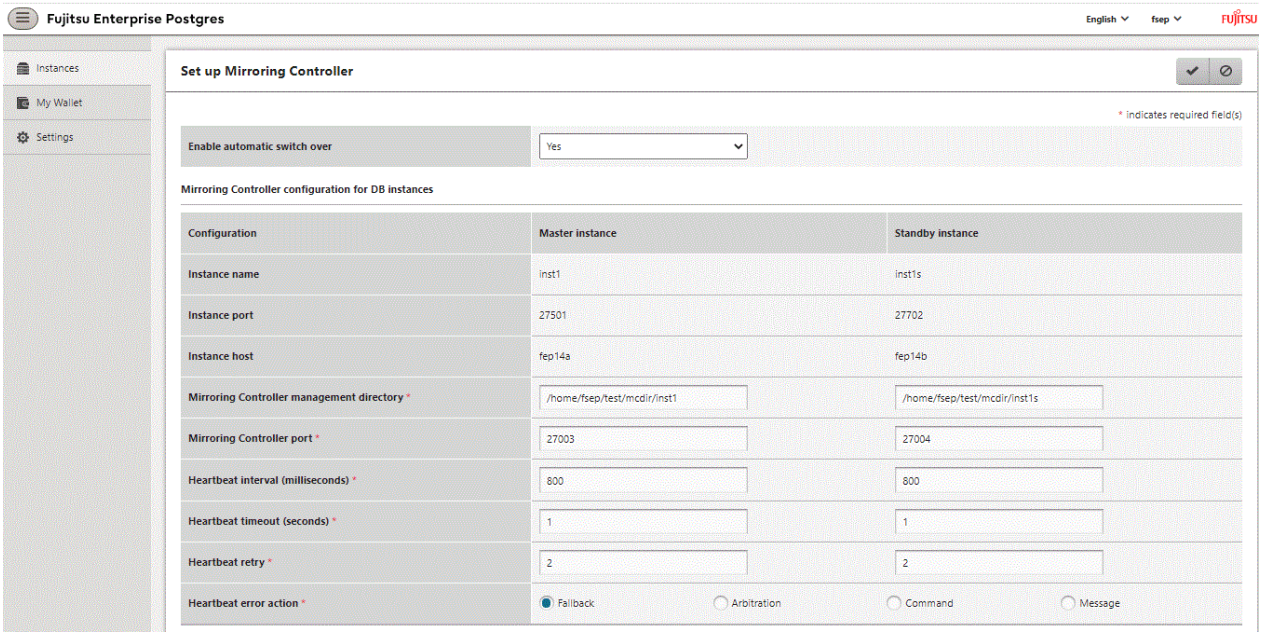
If you set up the arbitration server using WebAdmin, install WebAdmin on the arbitration server.

5.1 Mirroring Controller Setup

Perform the following procedure to set up Mirroring Controller in a streaming replication cluster.

1. In the [Instances] tab, select the standby instance on which Mirroring Controller needs to be set up.
2. Click .
3. Enter the information for the Mirroring Controller to be set up.

   In the example below, Mirroring Controller is being set up for the replication cluster having master instance "inst1" and standby instance "inst1s".



The instance name, host address and port of the master and standby instances are displayed for easy reference.

Enter the following items on master instance and on standby instance fields for Mirroring Controller setup, as shown in the above screenshot:

- [Enable automatic switch over]: Toggles the automatic switch/disconnection functionality. Select "Yes". The default is "No".
- [Mirroring Controller management directory]: Directory where the Mirroring Controller configuration files will be stored. When the [Mirroring Controller management directory] is entered, WebAdmin will search the Mirroring Controller configuration files in the entered directory based on the [Data storage path] of the corresponding DB instance. If Mirroring Controller configuration files are found, the Mirroring Controller fields will be auto filled.
- [Mirroring Controller port]: Port number of Mirroring Controller. Note that if the Windows firewall feature is enabled, you must enable the port number of Mirroring Controller. Refer to "E.2 Windows Firewall Settings" for details.
- [Heartbeat interval (milliseconds)]: Number of milliseconds between two consecutive heartbeat checks. The default is "800".
- [Heartbeat timeout (seconds)]: Number of seconds for the heartbeat timeout. The default is "1".
- [Heartbeat retry]: Number of retries for heartbeat monitoring, before failover occurs. The default is "2".
- [Heartbeat error action]: Operation when a heartbeat abnormality is detected. The default is "Fallback".

  When using FUJITSU Enterprise Postgres 10 or 11 instance which was created with FUJITSU Enterprise Postgres WebAdmin 10 or 11, the mode in the [Heartbeat error action] for Mirroring Controller setup cannot be changed.

  When setting up Mirroring Controller for FUJITSU Enterprise Postgres 9.5 and 9.6 instances, the [Heartbeat error action] is not supported and therefore is not displayed.

When the [Heartbeat error action] is set to "Arbitration", the following extra items are displayed:

- [Arbitration network IP address]: IP address of the arbitration network.
- [Mirroring Controller Arbitration port]: Port number of Mirroring Controller for communicating with the arbitration server.

The [Arbitration server configuration] section is also displayed with the following items. The [Arbitration server configuration] will not be auto filled.

- [Location]: Location of the arbitration server. "Local" or "Remote" can be selected depending on your configuration.

  If the arbitration server and WebAdmin server are located on the same server, you can select "Local" and the following items are displayed:

  - [Arbitration management directory]: Directory where the arbitration server configuration files will be stored.
  - [Arbitration server host or IP address]: Host name or IP address of the arbitration server.
  - [Arbitration process port]: Port number for the arbitration process.
  - [Fencing command]: Full path of the fencing command that fences a database server when an abnormality is detected.

  If "Remote" is set for the item, the items below are displayed in addition to the above items.

  - In the [Arbitration server configuration] section, [Operating system credential] is displayed where you can enter the following information:

    [User name]: User name to access the arbitration server.

    [Password]: Password to access the arbitration server.

  - In the [Remote WebAdmin for Arbitration server] section, the following items are displayed:

    [Remote WebAdmin address]: IP address of the remote WebAdmin installed on the arbitration server.

    [Remote WebAdmin port]: Port number for the WebAdmin installed on the arbitration server.

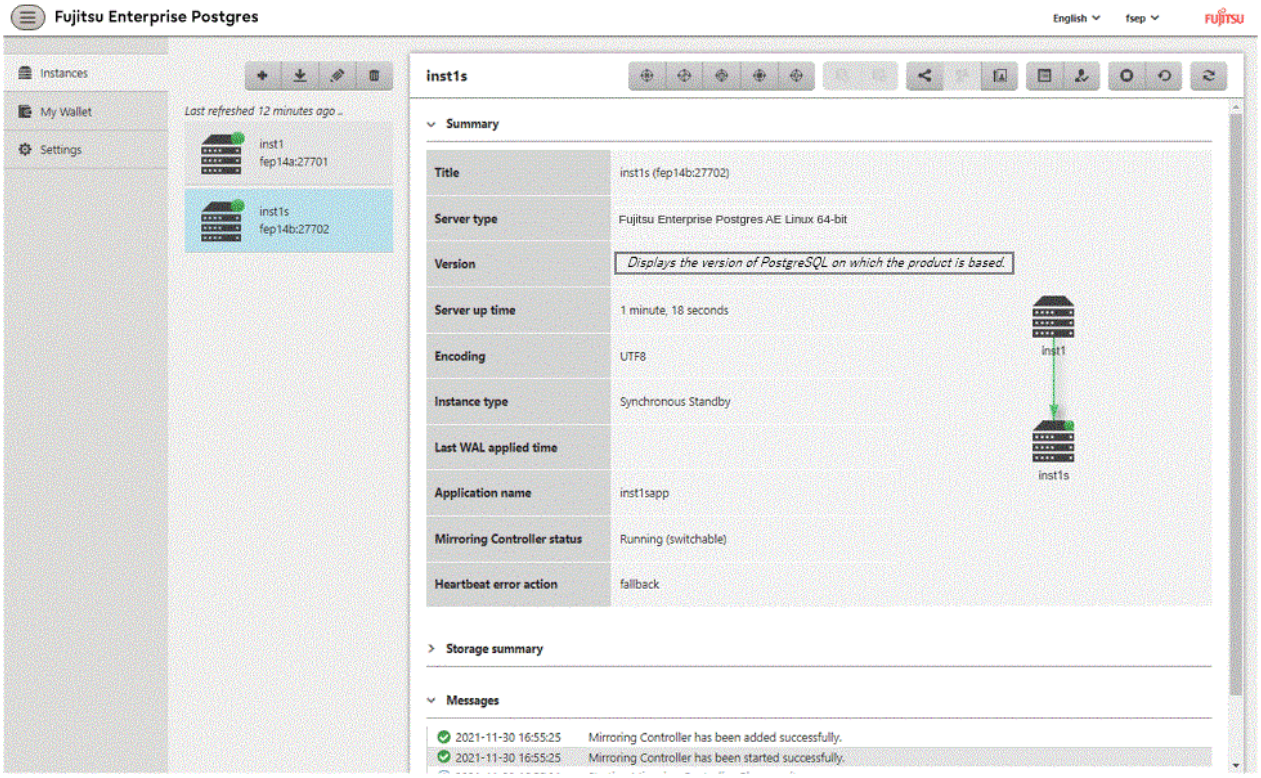
When the [Heartbeat error action] is set to "Command", the following extra items are displayed:

- [Arbitration command]: Full path of the arbitration command to be executed when an abnormality is detected.
- [Fencing command]: Full path of the fencing command that fences a database server when an abnormality is detected.

4. Click to set up Mirroring Controller.

5. Upon successful completion, Mirroring Controller will be started on master and standby instances.

   In the [Instances] tab, select standby instance. The page below is displayed, in which you can check the Mirroring Controller status. In the example below, standby instance "inst1s" is used.



After the Mirroring Controller has been set up, ([Edit Mirroring Controller] button) and ([Mirroring Controller Configuration] button) are available. These buttons are displayed only when FUJITSU Enterprise Postgres 10 or later instances are created with FUJITSU Enterprise Postgres WebAdmin 12 or later.

For FUJITSU Enterprise Postgres 9.5 and 9.6 instances, the [Heartbeat error action] will not be displayed.

When the [Heartbeat error action] is "Arbitration", the following information is displayed: whether the arbitration status is "online" or "offline", the arbitration server IP address and the arbitration process port.

Note

Operating system credential (User name, Password) should not contain hazardous characters. Refer to "Appendix F WebAdmin Disallow User Inputs Containing Hazardous Characters".

5.2 Edit Mirroring Controller Setup

Settings made in "5.1 Mirroring Controller Setup" can be updated in either the master instance or a standby instance using WebAdmin.

Perform the following procedure to edit Mirroring Controller configuration:

1. In the [Instances] tab, select the instance for which the Mirroring Controller configuration is to be edited.
2. Click .
3. Enter the information for the Mirroring Controller to be updated. Refer to "5.1 Mirroring Controller Setup".
4. Click to update the Mirroring Controller.
5. Upon successful completion, Mirroring Controller will be started on master and standby instances.

Editing and saving the [Edit Mirroring Controller] page will reset all other settings that are not listed on this page to default values.

5.3 Mirroring Controller Configuration

The information related to Mirroring Controller monitoring and control (refer to "A.4.1 Server Configuration File for the Database Servers") and the information related to arbitration and control of the Mirroring Controller arbitration process (refer to "A.4.2 Arbitration Configuration File") can be set using WebAdmin. You can view and update the configuration on either the master instance or the standby instance.

Perform the following procedure:

1. In the [Instances] tab, select the instance for the Mirroring Controller configuration you want to view.
2. Click to view the Mirroring Controller configuration.
3. Click to show the editing page for the Mirroring Controller configuration. The Mirroring Controller configurations defined during [Mirroring Controller Setup] are read-only on this page. Refer to "5.1 Mirroring Controller Setup".

   Additionally, refer to the "Appendix A Parameters" for information about the settings and the corresponding parameter names.

   The items common to all [Heartbeat error action] are:

   - Target DB
   - Core file path
   - Syslog facility
   - Syslog identity
   - Remote call timeout (milliseconds)
   - Agent alive timeout (seconds)
   - DB instance check interval (milliseconds)
   - DB instance check timeout (seconds)
   - DB instance check retry
   - DB instance timeout action
   - Disk check interval (milliseconds)
   - Disk check retry
   - Tablespace directory error action
   - Post-switch command
   - Post-promote command
     (Post-promote command is replaced in FUJITSU Enterprise Postgres 12. The Post-promote command is still valid and will be displayed when it is used in the server configuration file of Mirroring Controller.)
   - Post-attach command
   - Pre-detach command
   - State transition command timeout (seconds)
   - Check synchronous standby names validation

   When the [Heartbeat error action] is set to "Arbitration", the following extra items are displayed:

   - Arbitration timeout (seconds)
   - Arbiter alive interval (milliseconds)
   - Arbiter alive retry

- Arbiter alive timeout (seconds)
- Arbiter connect interval (milliseconds)
- Arbiter connect timeout (seconds)
- Fencing command
- Fencing command timeout (seconds)
- Shutdown detached synchronous standby

When the [Heartbeat error action] is set to "Arbitration", the [Arbitration server configuration] section is displayed with the following items:

- Core file path
- Syslog facility
- Syslog identity
- Fencing command timeout (seconds)
- Heartbeat interval (milliseconds)
- Heartbeat timeout (seconds)
- Heartbeat retry

When the [Heartbeat error action] is set to "Command", the following extra items are available:

- Fencing command timeout (seconds)
- Arbitration command timeout (seconds)
- Shutdown detached synchronous standby

When the [Heartbeat error action] is set to "Message", the following extra items are available:

- Fencing command
- Fencing command timeout (seconds)

In addition, the following configurations are provided:

- DB instance JDBC connection SSL parameters
- DB instance libpq connection SSL parameters

4. Click to update the Mirroring Controller configurations.

5.4 Stopping Mirroring Controller

Mirroring Controller can be stopped either in master instance or in standby instance using WebAdmin.

Perform the following procedure to stop Mirroring Controller.

1. In the [Instances] tab, select the instance where to stop Mirroring Controller.
2. Click .
3. In the confirmation dialog box, click [Yes].

Mirroring Controller will be stopped on the selected instance. The Mirroring Controller status will be updated, and a confirmation message entry will be displayed in the [Message] section.

5.5 Starting Mirroring Controller

Mirroring Controller can be started either in master instance or in standby instance using WebAdmin.

Perform the following procedure to start Mirroring Controller.

1. In the [Instances] tab, select the instance where to start Mirroring Controller.
2. Click .
3. In the confirmation dialog box, select the desired failover mode, and then click [Yes].

Mirroring Controller will be started on the selected instance. The Mirroring Controller status will be updated, and a confirmation message entry will be displayed in the [Message] section.

5.6 Disabling Failover Mode

Disabling failover mode in Mirroring Controller disables automatic switch/disconnection between master and standby instances.

Perform the following procedure to disable failover mode.

1. In the [Instances] tab, select the instance.
2. Click .
3. In the confirmation dialog box, click [Yes].

Failover mode will be disabled in Mirroring Controller. The Mirroring Controller status will be updated and a confirmation message entry will be displayed in the [Message] section.

5.7 Enabling Failover Mode

Enabling failover mode in Mirroring Controller enables automatic switch/disconnection between master and standby instances.

Perform the following procedure to enable failover.

1. In the [Instances] tab, select the instance.
2. Click .
3. In the confirmation dialog box, click [Yes].

Failover mode will be enabled in Mirroring Controller. The Mirroring Controller status will be updated and a confirmation message entry will be displayed in the [Message] section.

5.8 Deleting Mirroring Controller Setup

Deleting Mirroring Controller setup removes its setup from master and standby instances.

1. In the [Instances] tab, select the instance.
2. Click .
3. In the confirmation dialog box, click [Yes].

Mirroring Controller setup will be removed from the cluster. The cluster status will be updated and a confirmation message entry will be displayed in the [Message] section.

For the instances in FUJITSU Enterprise Postgres 12 or later, WebAdmin does not delete the Mirroring Controller management directory and the configuration files.

5.9 Status Update after Failover

When Mirroring Controller performs a failover, standby instance will be promoted to standalone instance. The Mirroring Controller setup will be removed from both standby and master instances.

The following scenario describes one of the ways in which failover can be triggered, and the results achieved by the use of Mirroring Controller in WebAdmin.

1. In the [Instances] tab, select the master instance "inst1".
2. Click .
3. In the confirmation dialog box, the warning "This instance is being monitored by Mirroring Controller. Stopping the instance may result in cluster failover." is displayed.
4. Choose the stop mode and click [Yes].

   In the server, the following takes place:

   a. The master instance is stopped.
   b. Failover is triggered in Mirroring Controller.
   c. The Mirroring Controller setup is removed from both master and standby instances
   d. Standby instance is promoted to standalone.
5. When the instance is refreshed in WebAdmin, the latest status of the instances will be displayed.

When failover is performed, the Mirroring Controller setup is removed from both master and standby instances. Therefore, to manage the Mirroring Controller using WebAdmin again, create the standby instance and set up Mirroring Controller.

Refer to "Creating a Standby Instance" in the Operation Guide for details.

Refer to "5.1 Mirroring Controller Setup" for details.

5.10 Action Required when an Error Occurs in the Combined Admin Network and Log Transfer Network

Communication errors may temporarily occur in the network used as the admin network and log transfer network due to reasons such as high load on the server or insufficient system resources. Because of this, there is a risk of causing a split-brain situation by mistake even though the server has no issues.

Split brain is a phenomenon in which both servers temporarily operate as primary servers, causing data updates to be performed on both servers.

How to detect split brain using WebAdmin

If the conditions below are met, split brain may occur. Refer to "Split-brain detection method" and "How to recover from a split-brain" in "Appendix D Notes on Performing Automatic Degradation Immediately after a Heartbeat Abnormality" and take the actions described.

1. A standby instance is selected in the [Instances] tab, and
2. "Standalone" is displayed in [Instance type], and
3. A master instance is selected in the [Instances] tab, and
4. "Standalone" is displayed in [Instance type].

The admin network is important because Mirroring Controllers use it to confirm the status of each server.

The log transfer network is also important to maintain the data freshness.

Therefore, use network configurations resistant to faults for these networks by using the network redundancy channel bonding feature provided by the operating system or network driver vendor.

5.11 Performing Automatic Degradation Using the Arbitration Server

If database multiplexing is performed using WebAdmin, it is also possible to perform automatic degradation using the arbitration server. In such cases, it is necessary to perform tasks on the database server and the arbitration server after setting up Mirroring Controller in WebAdmin.

Tasks on the arbitration server

Perform setup of the arbitration server using Mirroring Controller commands.

1. Set up the arbitration server.

   Refer to "2.3 Setting Up the Arbitration Server" in "Chapter 2 Setting Up Database Multiplexing Mode" for information on how to set up the arbitration server.

Tasks on the database server

Change some of the settings after setting up Mirroring Controller in WebAdmin.

1. Set up Mirroring Controller in WebAdmin.

   Refer to "5.1 Mirroring Controller Setup" for details.

2. Use WebAdmin to stop Mirroring Controller on the master and standby instances.

   Refer to "5.4 Stopping Mirroring Controller" for details.

3. Edit the network configuration file of the master and standby instances, and add the arbitration server information.

   The network configuration file is network.conf, which exists in the Mirroring Controller management directory specified during Mirroring Controller setup. Refer to "A.3 Network Configuration File" for details.

   A definition example of network.conf is shown below.

   ```
   Example:
   The port number of the database server to be used as the arbitration network is set to "27541".
   The ID of the server of the Mirroring Controller arbitration process is set to "arbiter", and its
   port number is set to "27541".
   ```

   ```
   dbsvm27500 192.0.2.100,192.0.3.100 27540,27541 server
   dbsvs27500 192.0.2.110,192.0.3.110 27540,27541 server
   arbiter 192.0.3.120 27541 arbiter
   ```

   Note

   - Ensure that the port numbers set for the database server and the arbitration server do not conflict with other software. In addition, do not configure the same segment for the admin network and the arbitration network.
   - If the server type is "server", two IP addresses or host names, and two port numbers need to be specified in the following order:
     - IP address or host name of the database server used as the admin network
     - IP address or host name of the database server used as the arbitration network
     - Port number of the database server used as the admin network
     - Port number of the database server used as the arbitration network

- If the server type is "arbiter", specify the IP address or host name set for the my_address parameter and the port number set for the port parameter in arbitration.conf of the arbitration server.
- WebAdmin also support editing mirroring controller configuration via Use WebAdmin to edit Mirroring Controller configurations.
  Refer to "5.2 Edit Mirroring Controller Setup" for detailes.

4. Edit the server configuration file of the master and standby instances, and add the parameters required for automatic degradation using the arbitration server.

   The server configuration file is *instanceName*.conf or *instancePort*.conf, which exists in the Mirroring Controller management directory specified during Mirroring Controller setup.

   To perform automatic degradation using the arbitration server, set the heartbeat_error_action parameter to "arbitration".

   Refer to "A.4.1 Server Configuration File for the Database Servers" for information on other parameters.

5. Use WebAdmin to start Mirroring Controller on the master and standby instances.

   Refer to "5.5 Starting Mirroring Controller" for details.

Common tasks

1. Use the Mirroring Controller command to check the connection status from the database server or the arbitration server.

   Refer to "2.8 Checking the Connection Status" for information on how to check the connection status.

Appendix A Parameters

This appendix describes the configuration files and parameters required by the database multiplexing mode.

See

Refer to "Server Configuration" in the PostgreSQL Documentation for information on the postgresql.conf file.

A.1 Parameters Set on the Primary Server

The content for the parameters set in the postgresql.conf file of the primary server is shown in the table below.

Table A.1 postgresql.conf file

Parameter	Value set	Explanation
wal_level	replica or logical	Specify the output level for the transaction log. Specify "logical" when logical decoding is also to be used.
max_wal_senders	2 or more	Specify "2" when building a Mirroring Controller cluster system. When additionally connecting asynchronous standby servers to the cluster system, add the number of simultaneous connections from these standby servers.
synchronous_standby_names	'*standbyServerName*'	Use single quotation marks (') to enclose the name that will identify the standby server. Any name can be specified. Do not change this parameter while Mirroring Controller is running. Do not specify multiple names to this parameter as the Mirroring Controller can manage only one standby server.
hot_standby	on	Specify whether queries can be run on the standby server. Specify "on".
wal_keep_size	WAL save size (megabytes)	If a delay exceeding the value set in this parameter occurs, the WAL segment required later by the primary server may be deleted. Additionally, if you stop a standby server (for maintenance, for example), consider the stop time and set a value that will not cause the WAL segment to be deleted. Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on estimating the WAL save size.
wal_log_hints	on	When using the pg_rewind command to recover a standby server, specify this parameter or enable checksums when executing the initdb command.
wal_sender_timeout	Timeout (milliseconds)	Specify the time period after which it is determined that the receiver process (walreceiver) of the transaction log is in an abnormal state on the primary server. The specified value must be larger than the value set for the wal_receiver_status_interval parameter set in the postgresql.conf file of the standby server.

Parameter	Value set	Explanation
		By aligning this value with the value for the database process heartbeat monitoring time, you can unify the time after which it is determined that an error has occurred.
wal_receiver_timeout	Timeout (milliseconds)	Specify the time period after which it is determined that an error has occurred when the transaction log was received on the standby server. By aligning this value with the value for the database process heartbeat monitoring time, you can unify the time after which it is determined that an error has occurred.
archive_mode	on	Specify the archive log mode.
archive_command	'*installDir*/bin/pgx_walcopy.cmd "%p" "*backupDataStorageDestinationDirectory*/archived_wal/%f"'	Specify the command and storage destination to save the transaction log.
backup_destination	Backup data storage destination directory	Specify the name of directory where to store the backup data. Set the permissions so that only the instance administrator user can access the specified directory. Specify the same full path on all servers, so that the backup data of other servers can be used to perform recovery.
listen_addresses	Primary server IP address, host name, or "*"	Specify the IP address or host name of the primary server. Specify the IP address or corresponding host name that will be used to connect to the log transfer network. The content specified is also used to allow connections from client applications. To receive the connection and the transaction log from any client or standby server, specify "*". Refer to "Connections and Authentication" in the PostgreSQL Documentation for details.
max_connections	Number of simultaneous client connections to the instance + superuser_reserved_connections value	The value specified is also used to restrict the number of connections from client applications and the number of connections for the management of instances. Refer to "When an Instance was Created with the initdb Command" in the Installation and Setup Guide for Server, and "Connections and Authentication" in the PostgreSQL Documentation, for details.
superuser_reserved_connections	Add the number of simultaneous executions of mc_ctl status (*1) + 2	Specify the number of connections reserved for connections from database superusers. Add the number of connections from Mirroring Controller processes. Also reflect the added value in the max_connections parameter.
restart_after_crash	off	If "on" is specified, or the default value is used for this parameter, behavior equivalent to restarting Fujitsu Enterprise Postgres, including crash recovery, will be performed when some server processes end abnormally. However, when database multiplexing monitoring is used, a failover will occur after an error is detected when some server processes end abnormally, and the restart of those server processes is forcibly stopped. Specify "off" to

Parameter	Value set	Explanation
		prevent behavior such as this from occurring for no apparent reason.
synchronous_commit	on or remote_apply	Specify up to what position WAL send is to be performed before transaction commit processing returns a normal termination response to a client. The recommended value is "on" or "remote_apply" to prevent data loss caused by operating system or server down immediately after a switch or switch.
recovery_target_timeline	latest	Specify "latest" so that the new standby server (original primary server) will follow the new primary server when a switch occurs. This parameter is required when the original primary server is incorporated as a new standby server after the primary server is switched.

*1: Number of simultaneous executions of the mc_ctl command in the status mode.

A.2 Parameters Set on the Standby Server

This section explains the content of the file and parameters set on the standby server. After editing postgresql.conf file, start the instance.

The content for the parameters specified in postgresql.conf file is shown in the table below.

Table A.2 postgresql.conf file

Parameter	Value set	Explanation
wal_level	replica or logical	Specify the output level for the transaction log. Specify "logical" when logical decoding is also to be used.
max_wal_senders	2 or more	Specify "2" when building a Mirroring Controller cluster system. When additionally connecting asynchronous standby servers to the cluster system, add the number of simultaneous connections from these standby servers.
synchronous_standby_names	'*primaryServerName*'	Use single quotation marks (') to enclose the name that will identify the primary server. Any name can be specified. This name will be required to rebuild the original primary server as the new standby server after the primary server was switched. Do not change this parameter while Mirroring Controller is running. Do not specify multiple names to this parameter as the Mirroring Controller can manage only one standby server.
hot_standby	on	Specify whether queries can be run on the standby server. Specify "on".
wal_keep_size	WAL save size (megabytes)	If a delay exceeding the value set in this parameter occurs, the WAL segment required later by the standby server may be deleted by the primary server.

Parameter	Value set	Explanation
		Additionally, if you stop a standby server (for maintenance, for example), consider the stop time and set a value that will not cause the WAL segment to be deleted. Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on estimating the WALsave size.
wal_log_hints	on	When using the pg_rewind command to recover a standby server, specify this parameter or enable checksums when executing the initdb command.
wal_sender_timeout	Timeout (milliseconds)	Specify the time period after which it is determined that the receiver process (walreceiver) of the transaction log is in an abnormal state on the primary server. The specified value must be larger than the value set for the wal_receiver_status_interval parameter set in the postgresql.conf file of the standby server. By aligning this value with the value for the database process heartbeat monitoring time, you can unify the time after which it is determined that an error has occurred.
wal_receiver_timeout	Timeout (milliseconds)	Specify the time period after which it is determined that an error has occurred when the transaction log was received on the standby server. By aligning this value with the value for the database process heartbeat monitoring time, you can unify the time after which it is determined that an error has occurred.
backup_destination	Backup data storage destination directory	Specify the name of the backup data storage directory. Set the permissions so that only the instance administrator user can access the specified directory. Specify the same full path on all servers so that the backup data of other servers can be used to perform recovery.
archive_mode	on	Specify the archive log mode.
archive_command	'*installDir*/bin/pgx_walcopy.cmd "%p" "*backupDataStorageDestinationDirectory*/archived_wal/%f"'	Specify the command and storage destination to save the transaction log.
listen_addresses	Standby server IP address, host name, or "*"	Specify the IP address or host name of the standby server. Specify the IP address or corresponding host name that will be used to connect to the log transfer network. The content specified is also used to allow connections from client applications. To receive the connection and the transaction log from any client or standby server, specify "*". Refer to "Connections and Authentication" in the PostgreSQL Documentation for details.

Parameter	Value set	Explanation
max_connections	Number of simultaneous client connections to the instance + superuser_reserved_connections value	The value specified is also used to restrict the number of connections from client applications and the number of connections for the management of instances. Refer to "When an Instance was Created with the initdb Command" in the Installation and Setup Guide for Server, and "Connections and Authentication" in the PostgreSQL Documentation, for details.
superuser_reserved_connections	Add the number of simultaneous executions of mc_ctl status (*1) + 2	Specify the number of connections reserved for connections from database superusers. Add the number of connections from Mirroring Controller processes. Also reflect the added value in the max_connections parameter.
restart_after_crash	off	If "on" is specified, or the default value is used for this parameter, behavior equivalent to restarting Fujitsu Enterprise Postgres, including crash recovery, will be performed when some server processes end abnormally. However, when database multiplexing monitoring is used, a failover will occur after an error is detected when some server processes end abnormally, and the restart of those server processes is forcibly stopped. Specify "off" to prevent behavior such as this from occurring for no apparent reason.
synchronous_commit	on or remote_apply	Specify up to what position WAL send is to be performed before transaction commit processing returns a normal termination response to a client. The recommended value is "on" or "remote_apply" to prevent data loss caused by operating system or server down immediately after a switch or switch.
primary_conninfo	'*streamingReplicationConnectionDestinationInfo*'	Use single quotation marks (') to enclose the connection destination information of the streaming replication. The default value of this parameter is automatically set to postgresql.auto.conf in the procedure to run pg_basebackup for instance setup.
recovery_target_timeline	latest	Specify "latest" so that the new standby server (original primary server) will follow the new primary server when a switch occurs. This parameter is required when the original primary server is incorporated as a new standby server after the primary server is switched.

A.3 Network Configuration File

This section explains the network configuration file (network.conf) to be defined individually for the database servers and the arbitration server. Define the same content on the primary server and standby server.

For database multiplexing mode, define the network configuration for the following in network.conf.

- Integration between Mirroring Controller processes
- Integration between a Mirroring Controller process and the Mirroring Controller arbitration process

Items to be defined in network.conf

Format:

```
serverIdentifier hostName[,hostName] portNum[,portNum] [serverType]
Or,
serverIdentifier ipAddr[,ipAddr] portNum[,portNum] [serverType]
```

Specify the server identifier, IP address or host name, port number, and server type, using a space as the delimiter.

The items are explained in the table below.

Table A.3 network.conf file

Item	Description
serverIdentifier	Specify any identifier for the server. The maximum length is 64 bytes. Use ASCII characters excluding spaces and number signs (#) to specify this parameter.
ipAddrOrHostName	Specify the IP address or its corresponding host name that will connect to the admin network that performs communication between the database servers, and to the arbitration network that performs communication between a database server and the arbitration server. When specifying two IP addresses or host names delimited by a comma, do not insert a space after the comma. Use ASCII characters excluding spaces to specify the host name.
portNum	A port number cannot be specified if it exceeds the range 0 to 65535. Ensure that the port number does not conflict with other software. Do not specify an ephemeral port that may temporarily be assigned by another program. Note that the value specified in this parameter must also be set in the services file. When specifying two port numbers delimited by a comma, do not insert a space after the comma.
serverType	Specify "server" for a database server ("server" can be omitted), or "arbiter" for the arbitration server.

Content to be defined on the database servers

This section explains the network.conf content to be defined on the database servers.

The content to be defined depends on the operation settings at the time a heartbeat abnormality is detected.

When automatic degradation by the arbitration server is selected

- Specify definitions related to the admin network and arbitration network.
- Specify the IP address or host name and port number according to the server type (database server or arbitration server) as shown in the table below.

Server type	IP address or host name		Port number	
	First	Second	First	Second
server	IP address or host name used as the admin network	IP address or host name used as the arbitration network (*1)	Port number used as the admin network	Port number used as the arbitration network (*1)
arbiter	IP address or host name of the arbitration server Specify the same value as that specified in the my_address parameter of arbitration.conf on the arbitration server.	Not required	Port number on the arbitration server Specify the same value as that specified in the port parameter of arbitration.conf on the arbitration server.	Not required

*1: This value can be omitted from definitions not related to the local server. If it is omitted, network.conf must be created on both the primary server and standby server.

Example)

IPv4

```
server1 192.0.2.100,192.0.3.100 27540,27541 server
server2 192.0.2.110,192.0.3.110 27540,27541 server
arbiter 192.0.3.120 27541 arbiter
```

IPv6

```
server1 2001:258:8404:1217:250:56ff:fea7:559f,2001:258:8404:1217:250:56ff:fea8:559f
27540,27541 server
server2 2001:258:8404:1217:250:56ff:fea7:55a0,2001:258:8404:1217:250:56ff:fea8:55a0
27540,27541 server
arbiter 2001:258:8404:1217:250:56ff:fea8:55a0 27541 arbiter
```

When operation other than automatic degradation by the arbitration server is selected

- Specify definitions related to the admin network.
- Define the same content on the primary server and standby server.
- Define lines for database servers only.
- Specify only one IP address or host name and port number.

IP address or host name		Port number	
First	Second	First	Second
IP address or host name to be used as the admin network	Not required	Port number used as the admin network	Not required

Example)

The literal space represents a space.

IPv4

```
server1 192.0.2.100 27540
server2 192.0.2.110 27540
```

IPv6

```
server1 2001:258:8404:1217:250:56ff:fea7:559f 27540
server2 2001:258:8404:1217:250:56ff:fea7:55a0 27540
```

Content to be defined on the arbitration server

This section explains the network.conf content to be defined on the arbitration server.

- Specify definitions related to the arbitration network.
- Define lines for database servers only.
- For the IP address or host name, specify the same value as the second IP address or host name specified in the database server line in network.conf of the database server.

- For the port number, specify the same value as the second port number specified in the database server line in network.conf of the database server.

Example)

The literal space represents a space.

IPv4

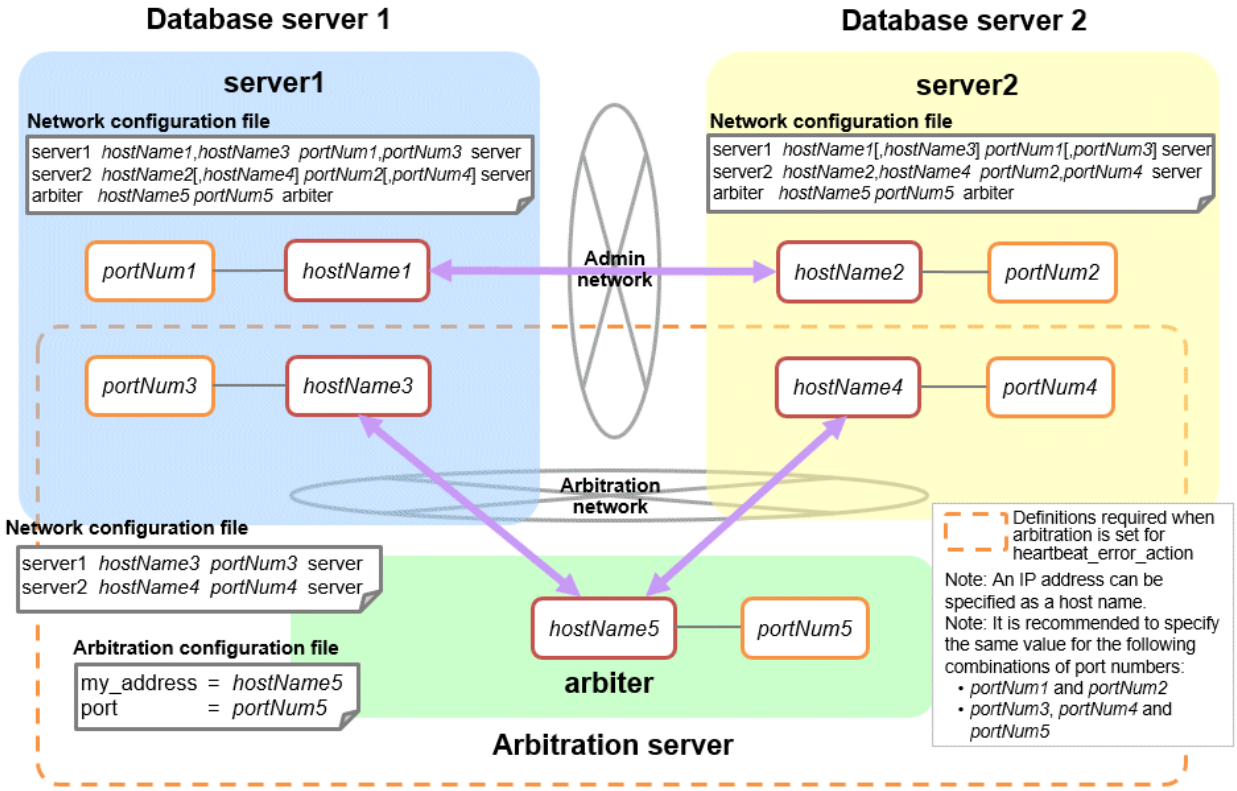
```
server1 192.0.3.100 27541
server2 192.0.3.110 27541
```

IPv6

```
server1 2001:258:8404:1217:250:56ff:fea8:559f 27541
server2 2001:258:8404:1217:250:56ff:fea8:55a0 27541
```

Relationship between network-related definitions

Refer to the diagram below for the relationship between the host names and IP addresses or port numbers specified in the network configuration file (network.conf) and arbitration configuration file (arbitration.conf).



A.4 Server Configuration File

A.4.1 Server Configuration File for the Database Servers

Define the information related to Mirroring Controller monitoring and control in the serverIdentifier.conf file. The maximum length of the server identifier is 64 bytes. Use ASCII characters excluding spaces to specify this parameter.

If the primary server and standby server environments are different, define content that is different, according to the environment.

Table A.4 serverIdentifier.conf file

Parameter	Value set	Explanation
db_instance	'*dataStorageDestinationDir*' [Example] db_instance = '/database1/inst1'	Specify using single quotation marks (') to enclose the data storage destination directory used to identify the monitoring target instance. Use ASCII characters to specify this parameter.
target_db	postgres or template1	Specify the name of the database to be connected to the database instance. The default is "postgres".
db_instance_username	'*usernameToConnectToDbInstance*'	Specify the username to connect to the database instance. Use ASCII characters to specify this parameter. Specify this parameter if the database administrator user is different from the operating system user who starts Mirroring Controller. Enclose the username of the database superuser in single quotation marks ('). The maximum length of the username is 63 bytes. The default is the operating system user who starts Mirroring Controller.
db_instance_password	'*passwordOfInstanceAdminUser*'	Specify the password used when Mirroring Controller connects to a database instance, enclosed in single quotation marks ('). Use ASCII characters to specify this parameter. If password authentication is performed, you must specify this parameter in the settings used when Mirroring Controller connects to a database instance. If you specify this parameter when password authentication is not performed, the parameter will be ignored. If the specified value of this parameter includes ' or \, write \' or \\, respectively.
enable_hash_in_password	on or off	Specify on to treat the # in the db_instance_password specification as a password character, or off to treat it as a comment. The default is "off".
core_file_path	*'coreFileOutputDir'*	Specify the directory to which the core file is to be output, enclosed in single quotation marks ('). Use ASCII characters to specify this parameter. If this parameter is omitted, it will be assumed that the Mirroring Controller management directory was specified.
syslog_facility	Specify LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, or LOCAL7.	When the import of logs to the syslog is enabled, the value of this parameter will be used for "facility" of the syslog. The default is "LOCAL0".
syslog_ident (*1)	'*programName*'	Specify using single quotation marks (') to enclose the program name used to identify the Mirroring Controller message in the system log.

Parameter	Value set	Explanation
		Use ASCII characters excluding spaces to specify this parameter. The default is 'MirroringControllerOpen'.
remote_call_timeout	Admin communication timeout	Specify the timeout value (milliseconds) of the Mirroring Controller agent process for communication between servers. Specify a value between 0 and 2147483647 to be less than the operation system TCP connection timeout (*2). In addition, when using the Mirroring Controller arbitrage process, fencing commands, and state transition commands, specify a value that is greater than the sum of the timeout values (*3). The value 0 indicates that there is no timeout limit. The default is 70000 milliseconds (70 seconds).
agent_alive_timeout	Timeout for Mirroring Controller process heartbeat monitoring (seconds)	If there is no response for at least the number of seconds specified, the Mirroring Controller process is restarted. Specify 0 or a value between 2 and 2147483647. The value 0 indicates that there is no timeout limit. The default is 0 seconds.
heartbeat_error_action	Operation when a heartbeat abnormality is detected using operating system or server heartbeat monitoring	arbitration: Perform automatic degradation using the arbitration server. command: Call a user exit to determine degradation, and perform automatic degradation if required. message: Notify messages. fallback: Perform automatic degradation unconditionally. The default is "arbitration". Set the same value on the primary server and standby server.
heartbeat_interval	Interval time for abnormality monitoring during heartbeat monitoring of the operating system or server (milliseconds)	Abnormality monitoring of the operating system or server is performed at the interval specified in heartbeat_interval. If an error is detected, operation will conform to the value specified for heartbeat_error_action. If "arbitration" is specified in heartbeat_error_action, the error detection time during monitoring of the operating system or server becomes longer than when the arbitration server is not used, by up to the value specified for arbitration_timeout. Specify a value between 1 and 2147483647. The specified value is used as the default for db_instance_check_interval and disk_check_interval. The default is 800 milliseconds.
heartbeat_timeout	Timeout for abnormality monitoring during heartbeat monitoring of the operating system or server (seconds)	If there is no response for at least the number of seconds specified, it will be assumed that an error has occurred that requires the primary server to be switched, or the standby server to be disconnected. If an error is detected, operation will conform to the value specified for heartbeat_error_action. If "arbitration" is

Parameter	Value set	Explanation
		specified in heartbeat_error_action, the error detection time during monitoring of the operating system or server becomes longer than when the arbitration server is not used, by up to the value specified for arbitration_timeout. Specify a value between 1 and 2147483647. The specified value is used as the default for db_instance_check_timeout. The default is 1 second.
heartbeat_retry	Number of retries for abnormality monitoring during heartbeat monitoring of the operating system or server (number of times)	Specify the number of retries to be performed when an error has been detected that requires the primary server to be switched, or the standby server to be disconnected. If an error is detected in succession more than the specified number of times, switch or disconnection will be performed. If an error is detected, operation will conform to the value specified for heartbeat_error_action. If "arbitration" is specified in heartbeat_error_action, the error detection time during monitoring of the operating system or server becomes longer than when the arbitration server is not used, by up to the value specified for arbitration_timeout. Specify a value between 0 and 2147483647. The specified value is used as the default for db_instance_check_retry and disk_check_retry. The default is 2 times.
db_instance_check_interval	Database process heartbeat monitoring interval (milliseconds)	Heartbeat monitoring of the database process is performed at the interval specified in db_instance_check_interval. This parameter setting is also used for abnormality monitoring of streaming replication. Specify a value between 1 and 2147483647. The default is the value set for heartbeat_interval.
db_instance_check_timeout	Database process heartbeat monitoring timeout (seconds)	If there is no response for at least the number of seconds specified, it will be assumed that an error has occurred that requires the primary server to be switched, or the standby server to be disconnected. Specify a value between 1 and 2147483647. The default is the value set for heartbeat_timeout.
db_instance_check_retry	Number of retries for database process heartbeat monitoring (number of times)	Specify the number of retries to be performed when an error has been detected that requires the primary server to be switched, or the standby server to be disconnected. If an error is detected in succession more than the specified number of times, switch or disconnection will be performed. However, if it detects that the database process is down, it will immediately switch or disconnect regardless of the setting of this parameter. This parameter setting is also used for abnormality monitoring of streaming replication. Specify a value between 0 and 2147483647.

Parameter	Value set	Explanation
		The default number of retries is the value set for heartbeat_retry.
db_instance_timeout_action	none, message, or failover	Specify the behavior for no-response monitoring of the instance. none: Do not perform no-response monitoring. message: Notify messages if an error is detected during no-response monitoring. failover: Perform automatic degradation if an error is detected during no-response monitoring. The default is "failover".
disk_check_interval	Interval time for disk abnormality monitoring (milliseconds)	Abnormality monitoring of disk failure is performed at the interval specified in disk_check_interval. If the file cannot be created, it will be assumed that an error has occurred that requires the primary server to be switched, or the standby server to be disconnected. Specify a value between 1 and 2147483647. The default is the value set for heartbeat_interval.
disk_check_retry	Number of retries for disk abnormality monitoring (number of times)	Specify the number of retries to be performed when an error has been detected that requires the primary server to be switched, or the standby server to be disconnected. If an error is detected in succession more than the specified number of times, switch or disconnection will be performed. Specify a value between 0 and 2147483647. The default number of retries is the value set for heartbeat_retry.
tablespace_directory_error_action	message or failover	Specify the behavior to be implemented if an error is detected in the tablespace storage directory. message: Notify messages. failover: Perform automatic degradation. The default is "failover".
arbiter_alive_interval	Interval time for monitoring connection to the Mirroring Controller arbitration process (milliseconds)	A heartbeat is sent to the Mirroring Controller arbitration process at the specified interval. Specify a value between 1 and 2147483647. The default is 16000 milliseconds. This parameter does not need to be set for operation that does not use the arbitration server.
arbiter_alive_timeout	Timeout for monitoring connection to the Mirroring Controller arbitration process (seconds)	If the heartbeat does not respond within the specified number of seconds, the Mirroring Controller arbitration process is determined to have been disconnected, a message is output, and reconnection is attempted. Specify a value between 1 and 2147483647. The default is 20 seconds. This parameter does not need to be set for operation that does not use the arbitration server.

Parameter	Value set	Explanation
arbiter_alive_retry	Number of retries for monitoring connection to the Mirroring Controller arbitration process (number of times)	Specify the number of heartbeat retries to be performed if an error is detected in the heartbeat to the Mirroring Controller arbitration process. If the heartbeat does not respond within the specified number of retries, the Mirroring Controller arbitration process is determined to have been disconnected. Specify a value between 0 and 2147483647. The default is 0 times. This parameter does not need to be set for operation that does not use the arbitration server.
arbiter_connect_interval	Attempt interval for connection to the Mirroring Controller arbitration process (milliseconds)	Reconnection is attempted at the specified interval if connection fails at startup of the Mirroring Controller process or if the Mirroring Controller arbitration process is disconnected. Specify a value between 1 and 2147483647. The default is 16000 milliseconds. This parameter does not need to be set for operation that does not use the arbitration server.
arbiter_connect_timeout	Timeout for connection to the Mirroring Controller arbitration process (seconds)	If reconnection at startup of the Mirroring Controller process or after disconnection of the Mirroring Controller arbitration process does not succeed within the specified number of seconds, connection to the Mirroring Controller arbitration process is determined to have failed and reconnection is attempted. Specify a value between 1 and 2147483647. The default is 20 seconds. This parameter does not need to be set for operation that does not use the arbitration server.
fencing_command	'*fencingCmdFilePath*' [Setting example] fencing_command = '/mc/fencing_dir/execute_fencing.sh'	Specify the full path of the fencing command that fences a database server where an error is determined to have occurred. Enclose the path in single quotation marks ('). Specify the path using less than 1024 bytes. This parameter must be specified when "command" is set for heartbeat_error_action.
fencing_command_timeout	Fencing command timeout (seconds)	If the command does not respond within the specified number of seconds, fencing is determined to have failed and a signal (SIGTERM) is sent to the fencing command execution process. Specify a value between 1 and 2147483647. The default is 20 seconds.
arbitration_timeout	Arbitration processing timeout in the Mirroring Controller arbitration process (seconds)	The specified value must be at least equal to the value of fencing_command_timeout in the arbitration configuration file, which is the heartbeat monitoring time of the operating system or server. If there is no response for at least the number of seconds specified, the primary server will not be switched and the

Parameter	Value set	Explanation
		standby server will not be disconnected. Therefore, perform degradation manually. If the heartbeat_interval, heartbeat_timeout, and heartbeat_retry values are specified in arbitration.conf for the arbitration server, use the arbitration server values to design arbitration_timeout. Specify a value between 1 and 2147483647. The default is 30 seconds. This parameter does not need to be set for operation that does not use the arbitration server.
arbitration_command	'*arbitrationCmdFilePath*' [Setting example] arbitration_command = '/mc/arbitration_dir/execute_arbitration_command.sh'	Specify the full path of the arbitration command to be executed when an abnormality is detected during heartbeat monitoring of the operating system or server. Enclose the path in single quotation marks ('). Specify the path using less than 1024 bytes. This parameter must be specified when "command" is set for heartbeat_error_action.
arbitration_command_timeout	Arbitration command timeout (seconds)	If the arbitration command does not respond within the specified number of seconds, it is determined that execution of the arbitration command has failed and a signal (SIGTERM) is sent to the arbitration command execution process. Specify a value between 1 and 2147483647. The default is 30 seconds. This parameter can be specified only when "command" is set for heartbeat_error_action.
shutdown_detached_synchronous_standby	on or off	Specify whether to forcibly stop the instance on the standby server when the standby server is disconnected. on: Stop the instance. off: Do not stop the instance. If "on" is specified and the pre-detach command was created, the pre-detach command is executed and then the instance is stopped. The default is "off".
post_switch_command	'*postSwitchCmdFilePath*' [Setting example] post_switch_command = '/mc/status_change/execute_post_switch.sh'	Specify the full path of the command to be called by Mirroring Controller after a new primary server is promoted during a failover of the primary server. Enclose the path in single quotation marks ('). Specify the path using less than 1024 bytes.
post_attach_command	'*postAttachCmdFilePath*' [Setting example] post_attach_command = '/mc/status_change/execute_post_attach.sh'	Specify the full path of the command to be called by Mirroring Controller after the standby server is attached to the cluster system. Enclose the path in single quotation marks ('). Specify the path using less than 1024 bytes.

Parameter	Value set	Explanation
pre_detach_command	'*preDetachCmdFilePath*' [Setting example] pre_detach_command = '/mc/status_change/execute_pre_detach.sh'	Specify the full path of the command to be called by Mirroring Controller before the standby server is disconnected from the cluster system. Enclose the path in single quotation marks ('). Specify the path using less than 1024 bytes.
status_change_command_timeout	State transition command timeout (seconds)	Specify the timeout value of the post-switch command, post-attach command, and pre-detach command. If the command does not respond within the specified number of seconds, a signal (SIGTERM) is sent to the execution process of the status change command. Specify a timeout between 1 and 2147483647. The default is 20 seconds.
check_synchronous_standby_names_validation	on or off	Specify whether Mirroring Controller is to periodically check during operations whether the synchronous_standby_names parameter in postgresql.conf was changed by an incorrect user operation. However, it is not recommended to enable this parameter, because performing this check causes Mirroring Controller to use the CPU of the database server redundantly and execute SQL statements at high frequency. This parameter is compatible with operations in FUJITSU Enterprise Postgres 9.6 or earlier. The default is "off".
db_instance_ext_pq_conninfo	*'libpqConnectionSSLParmToConnectToDbinstance'*	Specify, in key-value form, the connection parameter for libpq that Mirroring Controller adds when connecting to a database. The connection parameters you can specify are those related to SSL. Use ASCII characters to specify this parameter. The connection parameter specified in this parameter must also be specified in the db_instance_ext_jdbc_conninfo.
db_instance_ext_jdbc_conninfo	*'JDBCConnectionSSLParmToConnectToDbinstance'*	Specify, in URI form, the connection parameter for JDBC that Mirroring Controller adds when connecting to a database. The connection parameters you can specify are those related to SSL. Use ASCII characters to specify this parameter. The connection parameter specified in this parameter must also be specified in the db_instance_ext_pq_conninfo.

*1: By specifying the syslog_ident parameter of the postgresql.conf file, the Mirroring Controller output content can be referenced transparently, so log reference is easy.

*2: The operating system TCP connection timeout period is determined by the kernel parameter tcp_syn_retries. The remote_call_timeout parameter must be set to a value that is shorter than the timeout period for the operating system TCP connection timeout, so change either parameter as necessary.

*3: In management communications, arbitrage processing, fencing commands, and state transition commands may be executed in succession by the Mirroring Controller arbitrage process. Therefore, the value specified for the remote_call_timeout parameter must be greater than the sum of these timeout values. Depending on the value specified for the heartbeat_error_action parameter, set the remote_call_timeout parameter using the following formula:

- arbitration: (arbitration_timeout + fencing_command_timeout + status_change_command_timeout) * 1000
- command: (fencing_command_timeout + status_change_command_timeout) * 1000

- message: (fencing_command_timeout + status_change_command_timeout) * 1000
- fallback: (status_change_command_timeout) * 1000

Because other internal processing may be performed in the management communication, set the value obtained by multiplying the calculation result of the above equation by the safety factor (about 1.2).

Also, for the fencing_command_timeout parameter, use the value of the parameter in the database server's server definition file, not in the arbitration definition file.

The availability of some parameters depends on the value set for the heartbeat_error_action parameter that sets the operation to be performed if heartbeat monitoring of the operating system or server detects a heartbeat abnormality.

Table A.5 Parameter availability depending on the value set for the heartbeat_error_action parameter

Parameter	Value set			
	arbitration	command	message	fallback
arbiter_alive_interval	Y	N	N	N
arbiter_alive_timeout	Y	N	N	N
arbiter_alive_retry	Y	N	N	N
arbiter_connect_interval	Y	N	N	N
arbiter_connect_timeout	Y	N	N	N
arbitration_timeout	Y	N	N	N
arbitration_command	N	R	N	N
arbitration_command_timeout	N	Y	N	N
fencing_command	Y	R	Y	N
fencing_command_timeout	Y	Y	Y	N
shutdown_detached_synchronous_standby	Y	Y	N	N

R: Required

Y: Can be specified

N: Cannot be specified

A.4.2 Arbitration Configuration File

In arbitration.conf, define the information related to arbitration and control of the Mirroring Controller arbitration process.

Linux

Table A.6 arbitration.conf file (Linux)

Parameter	Value set	Description
port	Port number of the Mirroring Controller arbitration process	The specified value must not exceed the range 0 to 65535. Ensure that the port number does not conflict with other software. Do not specify an ephemeral port that may temporarily be assigned by another program. For the port number of the arbitration server to be specified in network.conf on the database server, specify the same value as the port number specified in this parameter.
my_address	'*ipAddrOrHostNameThatAcceptsConnectionFromMirroringControllerProcessOnDbServer*' [Setting example]	For the IP address or host name of the arbitration server to be specified in network.conf on the database server, specify the same value as the IP address or host name specified in this parameter.

Parameter	Value set	Description
	my_address = '192.0.3.120'	IPv4 and IPv6 addresses can be specified. Specify the IP address or host name, enclosed in single quotation marks (').
core_file_path	'*coreFileOutputDir*'	Specify the directory to which the core file is to be output, enclosed in single quotation marks ('). Use ASCII characters to specify this parameter. If this parameter is omitted, it will be assumed that the Mirroring Controller arbitration process management directory was specified.
syslog_facility	Specify LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, or LOCAL7.	When the import of logs to the syslog is enabled, the value of this parameter will be used for "facility" of the syslog. The default is "LOCAL0".
syslog_ident	'*programName*'	Specify using single quotation marks (') to enclose the program name used to identify the Mirroring Controller arbitration process message in the system log. Use ASCII characters excluding spaces to specify this parameter. The default is 'MirroringControllerArbiter'.
fencing_command	'*fencingCmdFilePath*' [Setting example] fencing_command = '/arbiter/fencing_dir/execute_fencing.sh'	Specify the full path of the fencing command that fences a database server where an error is determined to have occurred. Enclose the path in single quotation marks ('). Specify the path using less than 1024 bytes.
fencing_command_timeout	Fencing command timeout (seconds)	If the command does not respond within the specified number of seconds, fencing is determined to have failed and a signal (SIGTERM) is sent to the fencing command execution process. Specify a value between 1 and 2147483647. The default is 20 seconds.
heartbeat_interval(*1)	Interval time for heartbeat monitoring of the operating system or server (milliseconds)	The heartbeat monitoring of the database server is checked at the specified interval and arbitration is performed. Specify a value between 1 and 2147483647. The default is the value specified in *serverIdentifier*.conf of the database server. Specify this parameter to perform optimization taking into account differences in the line load to the admin network and the reduction in the time it takes to degrade.
heartbeat_timeout	Timeout for heartbeat monitoring of the operating system or server (seconds)	If there is no response for at least the number of seconds specified, it will be assumed that an error has occurred that requires the primary server or standby server to be fenced. Specify a value between 1 and 2147483647. The default is the value specified in *serverIdentifier*.conf of the database server.

Parameter	Value set	Description
		Specify this parameter to perform optimization taking into account differences in the line load to the admin network and the reduction in the time it takes to degrade.
heartbeat_retry	Number of retries for heartbeat monitoring of the operating system or server (number of times)	Specify the number of retries to be performed when an error has been detected that requires the primary server or standby server to be fenced. If an error is detected in succession more than the specified number of times, fencing will be performed. Specify a value between 0 and 2147483647. The default is the value specified in *serverIdentifier*.conf of the database server. Specify this parameter to perform optimization taking into account differences in the line load to the admin network and the reduction in the time it takes to degrade.

*1:Refer to "2.11.4 Tuning for Optimization of Degradation Using Abnormality Monitoring" for information on the tuning parameters for operating system or server abnormality monitoring when using an arbitration server.

Windows

Table A.7 arbitration.conf file (Windows)

Parameter	Value set	Description
port	Port number of the Mirroring Controller arbitration process	The specified value must not exceed the range 0 to 65535. Ensure that the port number does not conflict with other software. Do not specify an ephemeral port that may temporarily be assigned by another program. For the port number of the arbitration server to be specified in network.conf on the database server, specify the same value as the port number specified in this parameter.
my_address	'*ipAddrOrHostNameThatAcceptsConnectionFromMirroringControllerProcessOnDbServer*' [Setting example] my_address = '192.0.3.120'	For the IP address or host name of the arbitration server to be specified in network.conf on the database server, specify the same value as the IP address or host name specified in this parameter. IPv4 and IPv6 addresses can be specified. Specify the IP address or host name, enclosed in single quotation marks (').
core_file_path	'*coreFileOutputDir*'	Specify the directory to which the core file is to be output, enclosed in single quotation marks ('). Use ASCII characters, and specify "\\" as the path delimiter. If this parameter is omitted, it will be assumed that the Mirroring Controller arbitration process management directory was specified.
service_name	'*registeredServiceNameOfMirroringControllerArbitrationProcess*'	Specify the Mirroring Controller arbitration process service name to be registered as a Windows service, enclosed in single quotation

Parameter	Value set	Description
		marks ('). Use ASCII characters excluding forward slash (/) and backslash (\) to specify this parameter. The service name is up to 124 bytes. The default is 'MirroringControllerArbiter'.
event_source	*'eventSourceName'*	Specify the event source name to be used to identify the Mirroring Controller arbitration process message in the event log, enclosed in single quotation marks ('). Use ASCII characters to specify this parameter. The maximum length of the event source name is 255 bytes. The default is 'MirroringControllerArbiter'.
fencing_command	'*fencingCmdFilePath*' [Setting example] fencing_command = 'c:\\arbiter\\fencing_dir\\execute_fencing.bat'	Specify the full path of the fencing command that fences a database server where an error is determined to have occurred. Specify "\\" as the delimiter. Enclose the path in single quotation marks ('). Specify the path using less than 260 bytes. Any multibyte characters must use the same encoding as the operating system.
fencing_command_timeout	Fencing command timeout (seconds)	If the command does not respond within the specified number of seconds, fencing is determined to have failed and a signal (SIGTERM) is sent to the fencing command execution process. Specify a value between 1 and 2147483647. The default is 20 seconds.
heartbeat_interval(*1)	Interval time for heartbeat monitoring of the operating system or server (milliseconds)	The heartbeat monitoring of the database server is checked at the specified interval and arbitration is performed. Specify a value between 1 and 2147483647. The default is the value specified in *serverIdentifier*.conf of the database server. Specify this parameter to perform optimization taking into account differences in the line load to the admin network and the reduction in the time it takes to degrade.
heartbeat_timeout	Timeout for heartbeat monitoring of the operating system or server (seconds)	If there is no response for at least the number of seconds specified, it will be assumed that an error has occurred that requires the primary server or standby server to be fenced. Specify a value between 1 and 2147483647. The default is the value specified in *serverIdentifier*.conf of the database server. Specify this parameter to perform optimization taking into account differences in the line load to the admin network and the reduction in the time it takes to degrade.

Parameter	Value set	Description
heartbeat_retry	Number of retries for heartbeat monitoring of the operating system or server (number of times)	Specify the number of retries to be performed when an error has been detected that requires the primary server or standby server to be fenced. If an error is detected in succession more than the specified number of times, fencing will be performed. Specify a value between 0 and 2147483647. The default is the value specified in *serverIdentifier*.conf of the database server. Specify this parameter to perform optimization taking into account differences in the line load to the admin network and the reduction in the time it takes to degrade.

*1:Refer to "2.11.4 Tuning for Optimization of Degradation Using Abnormality Monitoring" for information on the tuning parameters for operating system or server abnormality monitoring when using an arbitration server.

Appendix B Supplementary Information on Building the Primary Server and Standby Server on the Same Server

The primary server and standby server can be pseudo-configured on the same server for system testing, for example. Out of consideration for performance and reliability, do not use this type of configuration for any other purposes. For this reason, do not use this type of configuration in a production environment.

Note that the setup and operations is the same as if the primary and standby servers are built on different servers.

This appendix provides supplementary information explaining how to configure the primary server and standby server on the same server.

Note

Even if automatic degradation by an arbitration server is set when the primary server and standby server are configured on the same server, there will be no effect of it.

B.1 Backup Data Storage Destination Directory

It is not a problem if the same backup data storage destination directory is used on the primary server and standby server.

B.2 How to Execute the mc_ctl Command

When executing the mc_ctl command, specify the server identifier in the --local-server option in order to identify the operation destination server.

Below is an example of starting Mirroring Controller of the server "server1" defined in the network.conf file. For mc_ctl command operations using another mode, also specify the --local-server option.

Define two server identifiers for the same IP address with different port numbers in the network.conf file.

```
Example)
```

```
server1 192.0.2.100 27540
server2 192.0.2.100 27541
```

Ensure that the port numbers of both primary server and standby server do not conflict with any other software.

Enabling automatic switch/disconnection

Start Mirroring Controller of the server "server1":

```
Example)
```

```
$ mc_ctl start -M /mcdir/inst1 --local-server server1
```

Stop Mirroring Controller of the server "server1":

```
Example)
```

```
$ mc_ctl stop -M /mcdir/inst1 --local-server server1
```

Disabling automatic switch/disconnection

Start Mirroring Controller of the server "server1":

```
Example)
```

```
$ mc_ctl start -M /mcdir/inst1 -F --local-server server1
```

Stop Mirroring Controller of the server "server1":

Example)

```
$ mc_ctl stop -M /mcdir/inst1 --local-server server1
```

Note

Add the --local-server option to the mc_ctl option specification for ExecStart and ExecStop of the unit file for systemd.

Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for details.

Appendix C User Commands

This appendix describes three categories of commands:

- Fencing command
- Arbitration command
- State transition commands

This appendix describes each category of user command.

C.1 Fencing Command

Format

The syntax for calling the fencing command from the Mirroring Controller process or the Mirroring Controller arbitration process is described below.

Fencing command of the database server

```
fencingCmd executionMode mcDegradationOper cmdServerId targetServerId primarycenter
```

Fencing command of the arbitration server

```
fencingCmd executionMode mcDegradationOper targetServerId
```

Input

Fencing command of the database server

Execution mode

monitor: Detect issues via automatic monitoring of the Mirroring Controller process

command: Mirroring Controller command execution (switch mode or detach mode of the mc_ctl command)

Degradation operation to be performed by Mirroring Controller

switch: Switch

detach: Disconnect

cmdServerId

ID of the database server that called the command

targetServerId

ID of the database server to be fenced

primarycenter

Fixed value

Fencing command of the arbitration server

Execution mode

monitor: Detect issues via automatic monitoring of the Mirroring Controller process

command: Mirroring Controller command execution (switch mode or detach mode of the mc_ctl command)

Degradation operation to be performed by Mirroring Controller

switch: Switch

detach: Disconnect

targetServerId

ID of the database server to be fenced

Output

Return value

0: Mirroring Controller will continue the degradation process.

Other than 0: Mirroring Controller will cancel the degradation process.

Description

Identifies the database server targeted for fencing based on the input server identifier, and implements the process that isolates it from the cluster system.

Notes

- The command is executed by the operating system user who started Mirroring Controller or the Mirroring Controller arbitration process. Therefore, if the command is to be executed by a specific operating system user, change the executing user of the command accordingly.
- The operating system user who started Mirroring Controller or the Mirroring Controller arbitration process must have execution privileges to the command. Otherwise, the degradation process will be canceled.
- From a security point of view, set the access privileges as necessary so that the fencing command is not overwritten and unauthorized operations are not performed by unintended operating system users.
- If the fencing command returns a value other than 0, Mirroring Controller will cancel the degradation process, so it is necessary for the user to check the status of the server, and switch or disconnect it manually.
- Before executing the fencing command, check if the server is already fenced, to avoid the command terminating abnormally.
- If the command times out, Mirroring Controller will stop the command, output an error message, and cancel the degradation process.

Information

The fencing command can be implemented by simply stopping the operating system or server. For example, if stopping the power for the database server, it is possible to use a utility to control the hardware control board in environments equipped with boards compatible with IPMI hardware standard.

Linux

Below is a sample script of a fencing command that powers off the database server using the IPMI tool.

Sample shell script

```
/installDir/share/mcarb_execute_fencing.sh.sample
```

Windows

Below is a sample script of a fencing command that powers off the database server using IPMIUTIL.

Sample shell script

```
installDir\share\mcarb_execute_fencing.bat.sample
```

C.2 Arbitration Command

Format

The syntax for calling the arbitration command from the Mirroring Controller process is described below.

```
arbitrationCmd cmdServerId targetServerId primarycenter
```

Input

cmdServerId

ID of the database server that called the command

targetServerId

ID of the database server to arbitrate

primarycenter

Fixed value

Output

Return value

0: The database server to arbitrate has an issue, and Mirroring Controller will continue the degradation process.

Other than 0: The database server to arbitrate is normal, and Mirroring Controller will cancel the degradation process.

Description

Identifies the database server to arbitrate based on the input server identifier, and checks the status of the server.

Notes

- The command is executed by the operating system user who started Mirroring Controller.
- The operating system user who started Mirroring Controller must have execution privileges to the command. Otherwise, the command will not be called, and the degradation process will be canceled.
- If the command times out, Mirroring Controller will stop the command, output an error message, and cancel the degradation process.

C.3 State Transition Commands

State transition commands include the three types of user commands below. Any of the commands can be implemented by Mirroring Controller in conjunction with database server status transitions.

- Post-switch command
- Pre-detach command
- Post-attach command

C.3.1 Post-switch Command

Format

The syntax for calling the post-switch command from the Mirroring Controller process is described below.

```
postswitchCmd serverIdentifier primarycenter
```

Input

serverIdentifier

ID of the database server (new primary server) that was switched

primarycenter

Fixed value

Output

Return value

None

Notes

- The command is executed by the operating system user who started Mirroring Controller.
- The operating system user who started Mirroring Controller must have execution privileges to the command. Otherwise, the command will not be called.
- If the command times out, Mirroring Controller will stop the command, output an error message, and cancel the process.

C.3.2 Pre-detach Command

Format

The syntax for calling the pre-detach command from the Mirroring Controller process is described below.

```
predetachCmd cmdServerId serverRole targetServerId primarycenter
```

Input

cmdServerId

ID of the database server that called the command

Server role

Role of the database server that called the command

primary: Primary

standby: Standby

targetServerId

ID of the standby server to be disconnected from the cluster system

primarycenter

Fixed value

Output

Return value

None

Notes

- The command is executed by the operating system user who started Mirroring Controller.
- The operating system user who started Mirroring Controller must have execution privileges to the command. Otherwise, the command will not be called, however, Mirroring Controller will output an error message and continue the process.
- If the command times out, Mirroring Controller will stop the command, output an error message, and cancel the process.

C.3.3 Post-attach Command

Format

The syntax for calling the post-attach command from the Mirroring Controller process is described below.

```
postattachCmd cmdServerId serverRole targetServerId primarycenter
```

Input

cmdServerId

ID of the database server that called the command

Server role

Role of the database server that called the command

primary: Primary

standby: Standby

targetServerId

ID of the standby server to be attached to the cluster system

primarycenter

Fixed value

Output

Return value

None

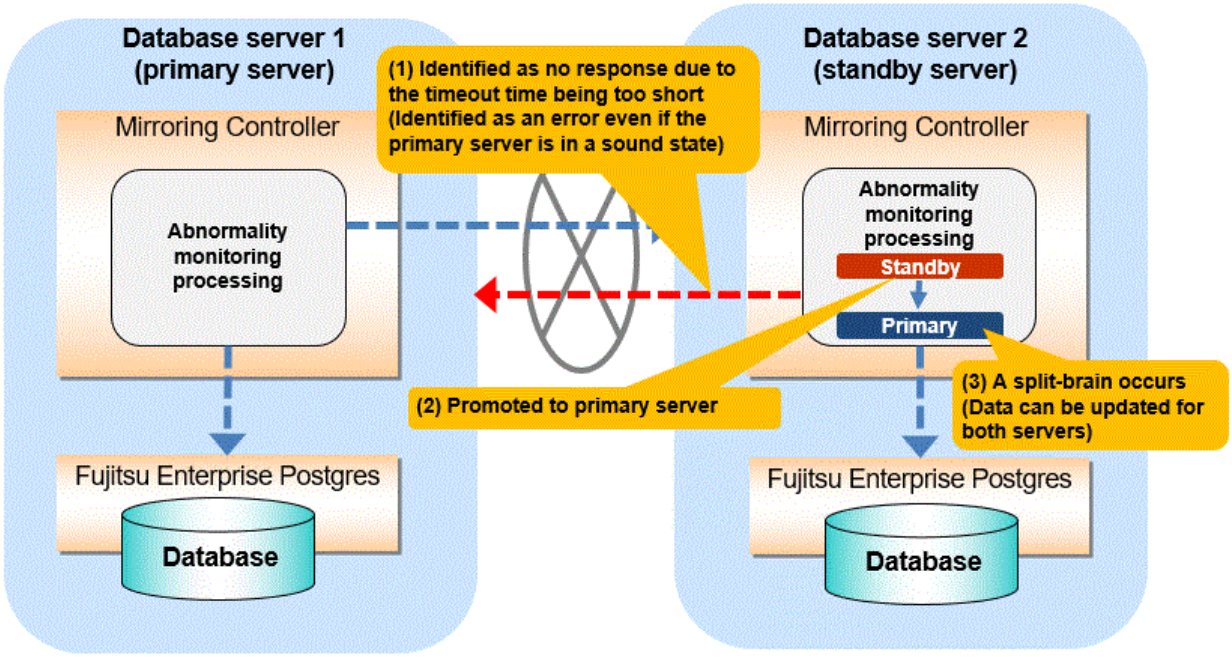
Notes

- The command is executed by the operating system user who started Mirroring Controller.
- The operating system user who started Mirroring Controller must have execution privileges to the command. Otherwise, the command will not be called.
- If the command times out, Mirroring Controller will stop the command, output an error message, and cancel the process.

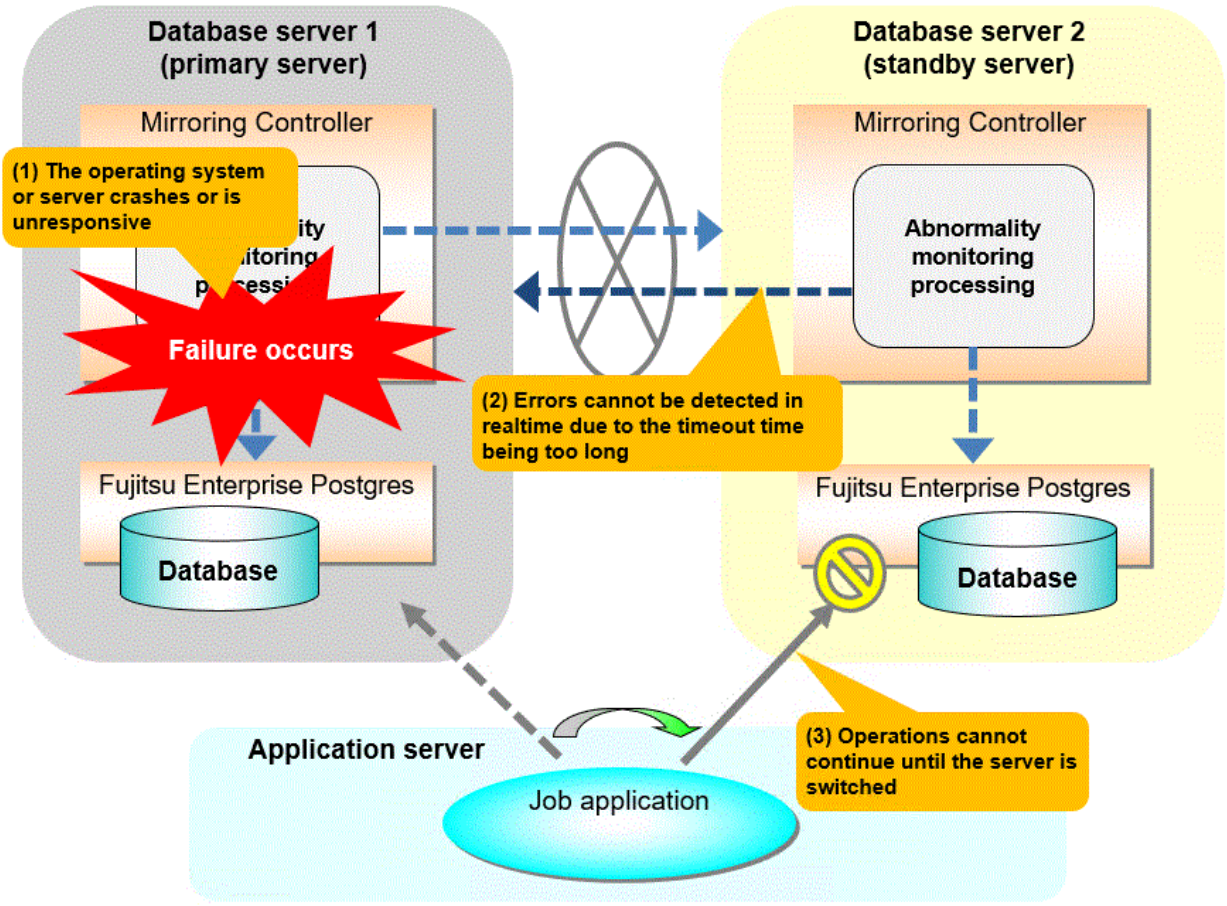
Appendix D Notes on Performing Automatic Degradation Immediately after a Heartbeat Abnormality

The type of issue below occurs if automatic degradation is performed unconditionally after an issue is detected during heartbeat monitoring of an operating system or server, and heartbeat monitoring was not properly tuned.

- **If the timeout time is too short**



● If the timeout time is too long



Notes on monitoring when the operating system or server crashes or is unresponsive

As illustrated in the diagram above, timeout is used to monitor whether the operating system or server crashes or is unresponsive. Therefore, if tuning has not been performed correctly, there is a risk of a split-brain mistakenly occurring even if the server is in a sound state.

Split-brain is a phenomenon in which both servers temporarily operate as primary servers, causing data updates to be performed on both servers.

Split-brain detection method

It can be confirmed that split-brain occurs under the following conditions:

1. When the mc_ctl command is executed in status mode on both servers, the "host_role" of both servers is output as "primary", and
2. The following message is output to the system log of one of the servers:

```
promotion processing completed (MCA00062)
```

How to recover from a split-brain

Use the procedure described below. Note that the new primary server is the server that was confirmed in step 2 of the aforementioned detection method.

1. Stop all applications that are running on the old and new primary servers.
2. Investigate and recover the database.
   Investigate the update results that have not been reflected to the new primary server from the database of the old primary server, and apply to the new primary server as necessary.
3. Stop the old primary server instance and the Mirroring Controller.
4. Resume the applications that were stopped in step 1.

5. Recover the old primary server.
   While referring to "2.5 Setting Up the Standby Server", build (set up) the old primary server as the new standby server, from the new primary server.

Appendix E Supplementary Procedure on Configuring for Operation in Database Multiplexing Mode

This appendix explains a supplementary procedure on the configuration required for operation in database multiplexing mode.

E.1 Security Policy Settings

This section explains how to configure the security settings to enable an operating system user account designated as an instance administrator user to log on as a service.

1. Displaying the [Local Security Policy] window

   In Windows, select [Administrative Tools], and then click [Local Security Policy].

2. Setting up security

   1. In the [Local Security Policy] window, select [Security Settings], select [Local Policies], and then click [User Rights Assignment].
   2. Under [Policy] in the [User Rights Assignment] window, double-click [Log on as a service].
   3. In the [Log on as a service Properties] window, set the following:
      a. Select the [Local Security Setting] tab.
      b. On the [Local Security Setting] tab, click [Add User or Group].
      c. In the [Select Users or Groups] window, enter the operating system user account of the instance administrator user in [Enter the object names to select].
      d. Click [OK].
   4. In the [Log on as a service Properties] window, click [OK].
   5. From the [Local Security Policy] tree, click [Local Policies], and then double-click [Security Options].
   6. Scroll down and double-click [User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode].
   7. From the drop-down menu, select the "Elevate without prompting" in the [Local Security Setting] tab.
   8. Click [OK].

E.2 Windows Firewall Settings

This section explains how to enable the port number used by Mirroring Controller, if the Windows firewall feature is enabled.

Windows Server(R) 2016:

1. In the [Windows Firewall] window, click [Advanced settings] on the left side of the window.
2. In the [Windows Firewall with Advanced Security] window, click [Inbound Rules] on the left side of the window.
3. Click [New Rule] on the right side of the window.
4. In the [New Inbound Rule Wizard] window, select [Port], and then click [Next].
5. Select [TCP] and [Specific local ports], then specify the port number defined in the network definition file, and then click [Next].
6. Select [Allow the connection], and then click [Next].
7. Select the profiles for which this rule applies, and then click [Next].
8. In [Name], specify the desired name, and then click [Finish].
9. In the [Windows Firewall with Advanced Security] window, check if the added rule is enabled under [Inbound Rules] in the center of the window.

Other than above:

1. In the [Windows Defender Firewall] window, click [Advanced settings] on the left side of the window.
2. In the [Windows Defender Firewall with Advanced Security] window, click [Inbound Rules] on the left side of the window.
3. Click [New Rule] on the right side of the window.
4. In the [New Inbound Rule Wizard] window, select [Port], and then click [Next].
5. Select [TCP] and [Specific local ports], then specify the port number defined in the network definition file, and then click [Next].
6. Select [Allow the connection], and then click [Next].
7. Select the profiles for which this rule applies, and then click [Next].
8. In [Name], specify the desired name, and then click [Finish].
9. In the [Windows Defender Firewall with Advanced Security] window, check if the added rule is enabled under [Inbound Rules] in the center of the window.

Appendix F WebAdmin Disallow User Inputs Containing Hazardous Characters

WebAdmin considers the following as hazardous characters, which are not allowed in user inputs.

```
| (pipe sign)

& (ampersand sign)

; (semicolon sign)

$ (dollar sign)

% (percent sign)

@ (at sign)

' (single apostrophe)

" (quotation mark)

\' (backslash-escaped apostrophe)

\" (backslash-escaped quotation mark)

<> (triangular parenthesis)

() (parenthesis)

+ (plus sign)

CR (Carriage return, ASCII 0x0d)

LF (Line feed, ASCII 0x0a)

, (comma sign)

\ (backslash)
```

Appendix G Collecting Failure Investigation Data

If the cause of an error that occurs while building the environment or during operations is unclear, data must be collected for initial investigation.

This appendix describes how to collect data for initial investigation.

Using a database server to collect data

- RHEL7,RHEL8,RHEL9 and SLES 12

  Use FJQSS (Information Collection Tool) to collect data for initial investigation.

- SLES 15

  Use the pgx_fjqssinf command to collect data for initial investigation.

Using an arbitration server to collect data

- Linux
  - RHEL7,RHEL8,RHEL9 and SLES 12

    Use FJQSS (Information Collection Tool) to collect data for initial investigation.

  - SLES 15

    Use the pgx_fjqssinf command to collect data for initial investigation.
- Windows

  Use FJQSS (Information Collection Tool) to collect data for initial investigation.

See

- Linux
  - Refer to the FJQSS manual for information on how to use FJQSS.
  - Refer to the Reference for information on the pgx_fjqssinf command.
- Windows

  Refer to the following manual for information on how to use FJQSS.
  - In the [Apps] menu, select [FJQSS (Information Collection Tool)], and then click [FJQSS User's Guide].

Note

Using a database server to collect data

- When using FJQSS to collect data for initial investigation, you must set the following environment variables:
  - Environment variable required for collecting data from the database

    Refer to "Collecting Failure Investigation Data" in the Operation Guide.
  - MCCONTROLDIR
  - Set the Mirroring Controller management directory.
    Refer to "1.2.1 Mirroring Controller Resources" for information on the Mirroring Controller management directory.

Using an arbitration server to collect data

Linux

- When using FJQSS to collect data for initial investigation, you must set the following environment variables:

  - ARBCONTROLDIR

    Set the Mirroring Controller arbitration process management directory.
    Refer to "1.2.1 Mirroring Controller Resources" for information on the Mirroring Controller arbitration server management directory.

  - ARBUSER

    Specify the operating system user who started the Mirroring Controller arbitration process.

  - FSEP_SA_HOME

    Set the Server Assistant installation directory.

Windows

- When using FJQSS to collect data for initial investigation, a window will be displayed for you to set the following environment variables:

  - ARBCONTROLDIR

    Set the Mirroring Controller arbitration process management directory.
    Refer to "1.2.1 Mirroring Controller Resources" for information on the Mirroring Controller arbitration server management directory.

Index

[A]

Action Required when a Heartbeat Abnormality is Detected.. 66
Action Required when All Database Servers or Instances Stopped 92
Action Required when an Error Occurs in the Database Multiplexing Mode..... 82
Action Required when Automatic Disconnection Fails..... 92
Action Required when Automatic Switch Fails..... 91
Action Required when Server Degradation Occurs..... 82
Addressing Errors During Degrading Operation..... 90
Application Connection Server Settings..... 40
arbitration.conf file (Linux)..... 123
arbitration.conf file (Windows)..... 125
Arbitration Command..... 131
Arbitration Configuration File..... 123
Arbitration Server Maintenance..... 74
Arbitration Server Process..... 10
Arbitration Server Resources..... 10
Authentication of the Standby Server..... 14

[B]

Backing up Database Multiplexing Mode Information..... 58
Backup Data Storage Destination Directory..... 128
Backup Operation..... 58

[C]

Changes in Operation..... 75
Changes Required when the Standby Server is Stopped..... 75
Changing from Database Multiplexing Mode to Single Server Mode..... 77
Changing from Single Server Mode to Database Multiplexing Mode..... 76
Changing Parameters..... 81
Changing to Database Multiplexing Mode when the Arbitration Server is Used for Automatic Degradation..... 80
Checking the Behavior..... 40
Checking the Connection Status..... 38
Checking the Connection Status on a Database Server..... 38
Checking the Connection Status on the Arbitration Server..... 39
Checking the Database Multiplexing Mode Status..... 63
Checking the Status of the Arbitration Server..... 64
Checking the Status of the Database Server..... 63
Configuring ICMP..... 15
Configuring the Arbitration Server..... 18
Confirming the Streaming Replication Status..... 37
Creating, Setting, and Registering the Primary Server Instance 28
Creating, Setting, and Registering the Standby Server Instance 34
Creating Applications..... 40

[D]

Database Backup Operation..... 58
Database Server Processes..... 10
Deciding on Operation when a Heartbeat Abnormality is Detected..... 11

[E]

Encryption of Transaction Logs Transferred to the Standby Server..... 14

[F]

Failback of the Primary Server..... 87
Fencing Command..... 130

[H]

How to Execute the mc_ctl Command..... 128

[I]

Identify cause of error and perform recovery..... 85,90
Identify Cause of Error and Restore the Standby Server..... 84
Identify Cause of Error and Restructure the Standby Server... 89
If Performing the Referencing Job on the Synchronous Standby Server..... 6
If Prioritizing the Main Job on the Primary Server..... 6
Installation..... 16

[M]

Manually Disconnecting the Standby Server..... 65
Manually Switching the Primary Server..... 65
Matching the system times..... 15
Mirroring Controller Resources..... 9
Monitoring Mirroring Controller Messages..... 66
Monitoring Using Database Multiplexing Mode..... 4

[N]

network.conf file..... 113
Network Configuration File..... 112
Notes on CPU Architecture and Products..... 11
Notes on Performing Automatic Degradation Immediately after a Heartbeat Abnormality..... 135

[O]

Operations in Database Multiplexing Mode..... 60
Operations when the Server has Started Degrading after a Disconnection has Occurred..... 88
Operations when the Server has Started Degrading after a Switch has Occurred..... 82
Overview of Database Multiplexing Mode..... 1

[P]

Parameters..... 108
Parameters Set on the Primary Server..... 108
Parameters Set on the Standby Server..... 110
Post-attach Command..... 133
Post-switch Command..... 132
postgresql.conf file..... 108,110
Pre-detach Command..... 133
Preparing for Setup..... 17
Preparing the Arbitration Server..... 17
Preparing the Backup Disk..... 17
Preparing the Database Server..... 17
Preparing to Output Error Logs to the Event Log (Windows). 17

[R]
Rebuild the Standby Server..... 87,90
Recovering from an Incorrect User Operation..... 96
Recovery of the Mirroring Controller management directory 85,90
Redundancy of the Admin and Log Transfer Networks.....11
Referencing on the Standby Server..... 6
Rolling Updates..... 69

[S]
Security in Database Multiplexing..... 12
Security Policy Settings.....138
Security Policy Settings (Windows).....18
ServerConfiguration File..... 115
Server Configuration File for the Database Servers.....115
serverIdentifier.conf file..... 116
Server Maintenance..... 68
Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances.....54
Setting Automatic Start and Stop of the Mirroring Controller Arbitration Process..... 56
Setting Up Database Multiplexing Mode..... 15
Setting Up Database Multiplexing Mode on the Primary Server24
Setting Up Database Multiplexing Mode on the Standby Server33
Setting Up the Arbitration Server.....18
Setting Up the Primary Server.....24
Setting Up the Standby Server.....33
Setup..... 15
Starting and Stopping Mirroring Controller..... 61
Starting and Stopping the Mirroring Controller Arbitration Process..... 60
Starting Mirroring Controller on the Primary Server..... 32
Starting Mirroring Controller on the Standby Server.....35
Starting the Mirroring Controller Arbitration Process..... 24,60
State Transition Commands.....132
Stop Mirroring Controller.....84,89
Stopping for Maintenance..... 74
Stopping the Mirroring Controller Arbitration Process.....60
Supplementary Information on Building the Primary Server and Standby Server on the Same Server..... 128
Supplementary Procedure on Configuring for Operation in Database Multiplexing Mode..... 138
System Configuration for Database Multiplexing Mode..... 6

[T]
Tuning.....40
Tuning for Optimization of Degrading Operation Using Abnormality Monitoring.....41
Tuning to Stabilize Queries on the Standby Server.....40
Tuning to Stabilize Queries on the Standby Server (when Performing Frequent Updates on the Primary Server)..... 41
Tuning to Stabilize the Database Multiplexing Mode.....40

[U]
Uninstalling in Database Multiplexing Mode..... 81
Users who perform setup and operations on the arbitration server 15
Users who perform setup and operations on the database server15

[W]
What is Database Multiplexing Mode..... 1
Windows Firewall Settings.....138

Fujitsu Enterprise Postgres 15 for x86

Cluster Operation Guide

Database Multiplexing

Windows

FUJITSU

B1WS-1474-01ENZ0(00)
April-2023

Preface

Purpose of this document

This document describes the tasks required for using the database multiplexing feature of Fujitsu Enterprise Postgres.

Intended readers

This document is intended for those who set up and use the database multiplexing feature.

Readers of this document are also assumed to have general knowledge of:

- PostgreSQL
- SQL
- Windows

Structure of this document

This document is structured as follows:

Chapter 1 Overview of Database Multiplexing Mode

Provides an overview of database multiplexing mode.

Chapter 2 Setting Up Database Multiplexing Mode

Describes how to set up database multiplexing mode.

Chapter 3 Operations in Database Multiplexing Mode

Explains periodic database multiplexing mode.

Chapter 4 Action Required when an Error Occurs in Database Multiplexing Mode

Explains the action required when an error occurs during a database multiplexing mode.

Chapter 5 Managing Mirroring Controller Using WebAdmin

Explains how to set up and manage Mirroring Controller in a streaming replication cluster using WebAdmin.

Appendix A Parameters

Explains the configuration files and parameters required for database multiplexing mode.

Appendix B Supplementary Information on Building the Primary Server and Standby Server on the Same Server

Explains supplementary information on building the primary server and standby server on the same server.

Appendix C User Commands

Explains the user commands.

Appendix D Notes on Performing Automatic Degradation Immediately after a Heartbeat Abnormality

Provides notes when performing automatic degradation unconditionally after a heartbeat abnormality is detected during heartbeat monitoring of an operating system or server.

Appendix E Supplementary Procedure on Configuring for Operation in Database Multiplexing Mode

Explains supplementary procedure on the configuration required for operation in database multiplexing mode.

Appendix F WebAdmin Disallow User Inputs Containing Hazardous Characters

Explains characters not allowed in WebAdmin.

Appendix G Collecting Failure Investigation Data

Explains how to collect data for initial investigation.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

```
Edition 1.0: April 2023
```

Copyright

Copyright 2016-2023 Fujitsu Limited

Contents

Chapter 1 Overview of Database Multiplexing Mode 1
1.1 What is Database Multiplexing Mode 1
1.1.1 Monitoring Using Database Multiplexing Mode 4
1.1.2 Referencing on the Standby Server 6
1.1.2.1 If Prioritizing the Main Job on the Primary Server 6
1.1.2.2 If Performing the Referencing Job on the Synchronous Standby Server 6
1.2 System Configuration for Database Multiplexing Mode 6
1.2.1 Mirroring Controller Resources 9
1.2.1.1 Database Server Resources 9
1.2.1.2 Arbitration Server Resources 10
1.2.2 Mirroring Controller Processes 10
1.2.2.1 Database Server Processes 10
1.2.2.2 Arbitration Server Process 10
1.2.3 Redundancy of the Admin and Log Transfer Networks 11
1.2.4 Notes on CPU Architecture and Products 11
1.3 Deciding on Operation when a Heartbeat Abnormality is Detected 11
1.4 Security in Database Multiplexing 12
1.4.1 Authentication of the Standby Server 14
1.4.2 Encryption of Transaction Logs Transferred to the Standby Server 15

Chapter 2 Setting Up Database Multiplexing Mode 16
2.1 Installation 17
2.2 Preparing for Setup 18
2.2.1 Preparing the Database Server 18
2.2.1.1 Preparing the Backup Disk 18
2.2.1.2 Preparatory Tasks for the Output of Error Logs to the Event Log 18
2.2.1.3 Security Policy Settings 19
2.2.2 Preparing the Arbitration Server 19
2.2.2.1 Preparing to Output Error Logs to the Event Log (Windows) 19
2.2.2.2 Security Policy Settings (Windows) 20
2.3 Setting Up the Arbitration Server 20
2.3.1 Configuring the Arbitration Server 20
2.3.2 Creating a User Exit for the Arbitration Server 25
2.3.3 Starting the Mirroring Controller Arbitration Process 25
2.4 Setting Up the Primary Server 26
2.4.1 Setting Up Database Multiplexing Mode on the Primary Server 26
2.4.2 Creating, Setting, and Registering the Primary Server Instance 31
2.4.3 Starting Mirroring Controller on the Primary Server 35
2.5 Setting Up the Standby Server 36
2.5.1 Setting Up Database Multiplexing Mode on the Standby Server 36
2.5.2 Creating, Setting, and Registering the Standby Server Instance 38
2.5.3 Starting Mirroring Controller on the Standby Server 40
2.6 Creating a User Exit for a Database Server 41
2.7 Confirming the Streaming Replication Status 42
2.8 Checking the Connection Status 43
2.8.1 Checking the Connection Status on a Database Server 43
2.8.2 Checking the Connection Status on the Arbitration Server 44
2.9 Creating Applications 45
2.9.1 Application Connection Server Settings 45
2.10 Checking the Behavior 45
2.11 Tuning 45
2.11.1 Tuning to Stabilize the Database Multiplexing Mode 45
2.11.2 Tuning to Stabilize Queries on the Standby Server 45
2.11.3 Tuning to Stabilize Queries on the Standby Server (when Performing Frequent Updates on the Primary Server) 46
2.11.4 Tuning for Optimization of Degradation Using Abnormality Monitoring 46

2.11.4.1 Tuning for Abnormality Monitoring of the Operating System or Server....46
2.11.4.1.1 Tuning Abnormality Monitoring for Operations that Use an Arbitration Server for Automatic Degeneration....47
2.11.4.1.2 Tuning Abnormality Monitoring for Operations that Perform Automatic Degeneration by Calling a User Exit that Determines Degeneration....52
2.11.4.1.3 Tuning Abnormality Monitoring for Operations that Notify Messages....54
2.11.4.1.4 Tuning Abnormality Monitoring for Operations that Perform Automatic Degenerate Unconditionally due to Heartbeat Abnormality....54
2.11.4.2 Tuning for Abnormality Monitoring of Darabase Processes....55
2.11.4.3 Tuning for Abnormality Monitoring of Streaming Replication....56
2.11.4.4 Tuning for Disk Abnormality Monitoring....58
2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances....59
2.13 Setting Automatic Start and Stop of the Mirroring Controller Arbitration Process....60
2.14 Backup Operation....62
2.14.1 Backing up Database Multiplexing Mode Information....62
2.14.2 Database Backup Operation....62

Chapter 3 Operations in Database Multiplexing Mode....64
3.1 Starting and Stopping the Mirroring Controller Arbitration Process....64
3.1.1 Starting the Mirroring Controller Arbitration Process....64
3.1.2 Stopping the Mirroring Controller Arbitration Process....64
3.2 Starting and Stopping Mirroring Controller....65
3.2.1 Starting Mirroring Controller....65
3.2.2 Stopping Mirroring Controller....67
3.3 Checking the Database Multiplexing Mode Status....68
3.3.1 Checking the Status of the Database Server....68
3.3.2 Checking the Status of the Arbitration Server....69
3.4 Manually Switching the Primary Server....69
3.5 Manually Disconnecting the Standby Server....70
3.6 Action Required when a Heartbeat Abnormality is Detected....70
3.7 Monitoring Mirroring Controller Messages....71
3.8 Server Maintenance....73
3.8.1 Rolling Updates....73
3.8.2 Stopping for Maintenance....78
3.8.3 Arbitration Server Maintenance....79
3.9 Changes in Operation....80
3.9.1 Changes Required when the Standby Server is Stopped....80
3.9.2 Changing from Single Server Mode to Database Multiplexing Mode....81
3.9.3 Changing from Database Multiplexing Mode to Single Server Mode....82
3.9.4 Changing to Database Multiplexing Mode when the Arbitration Server is Used for Automatic Degradation....85
3.9.5 Changing Parameters....86
3.9.6 Uninstalling in Database Multiplexing Mode....86

Chapter 4 Action Required when an Error Occurs in Database Multiplexing Mode....88
4.1 Action Required when Server Degradation Occurs....88
4.1.1 Operations when the Server has Started Degrading after a Switch has Occurred....88
4.1.1.1 Identify Cause of Error and Restore the Standby Server....90
4.1.1.1.1 Stop Mirroring Controller....90
4.1.1.1.2 Recovery of the Mirroring Controller management directory....91
4.1.1.1.3 Identify cause of error and perform recovery....91
4.1.1.2 Rebuild the Standby Server....93
4.1.1.3 Failback of the Primary Server....93
4.1.2 Operations when the Server has Started Degrading after a Disconnection has Occurred....94
4.1.2.1 Identify Cause of Error and Restore the Standby Server....95
4.1.2.1.1 Stop Mirroring Controller....95
4.1.2.1.2 Recovery of the Mirroring Controller management directory....96
4.1.2.1.3 Identify cause of error and perform recovery....96
4.1.2.2 Rebuild the Standby Server....96
4.1.3 Addressing Errors During Degrading Operation....96

4.2 Action Required when Automatic Switch Fails ... 97
4.3 Action Required when Automatic Disconnection Fails ... 97
4.4 Action Required when All Database Servers or Instances Stopped ... 98
4.5 Recovering from an Incorrect User Operation ... 101

Chapter 5 Managing Mirroring Controller Using WebAdmin ... 103
5.1 Mirroring Controller Setup ... 104
5.2 Edit Mirroring Controller Setup ... 106
5.3 Mirroring Controller Configuration ... 107
5.4 Stopping Mirroring Controller ... 108
5.5 Starting Mirroring Controller ... 108
5.6 Disabling Failover Mode ... 109
5.7 Enabling Failover Mode ... 109
5.8 Deleting Mirroring Controller Setup ... 109
5.9 Status Update after Failover ... 109
5.10 Action Required when an Error Occurs in the Combined Admin Network and Log Transfer Network ... 110
5.11 Performing Automatic Degradation Using the Arbitration Server ... 111

Appendix A Parameters ... 113
A.1 Parameters Set on the Primary Server ... 113
A.2 Parameters Set on the Standby Server ... 115
A.3 Network Configuration File ... 117
A.4 Server Configuration File ... 120
A.4.1 Server Configuration File for the Database Servers ... 120
A.4.2 Arbitration Configuration File ... 129

Appendix B Supplementary Information on Building the Primary Server and Standby Server on the Same Server ... 133
B.1 Backup Data Storage Destination Directory ... 133
B.2 Registering Service Names and Event Source Names in the Windows Service ... 133
B.3 How to Execute the mc_ctl Command ... 133

Appendix C User Commands ... 135
C.1 Fencing Command ... 135
C.2 Arbitration Command ... 136
C.3 State Transition Commands ... 137
C.3.1 Post-switch Command ... 137
C.3.2 Pre-detach Command ... 138
C.3.3 Post-attach Command ... 138

Appendix D Notes on Performing Automatic Degradation Immediately after a Heartbeat Abnormality ... 140

Appendix E Supplementary Procedure on Configuring for Operation in Database Multiplexing Mode ... 143
E.1 Security Policy Settings ... 143
E.2 Windows Firewall Settings ... 143

Appendix F WebAdmin Disallow User Inputs Containing Hazardous Characters ... 145

Appendix G Collecting Failure Investigation Data ... 146

Index ... 148

Chapter 1 Overview of Database Multiplexing Mode

This chapter provides an overview of database multiplexing mode.

Point

In this and subsequent chapters, the word "Mirroring Controller" may be used in the process or management directory name or explanation.

1.1 What is Database Multiplexing Mode

Database multiplexing mode is an operation mode (log shipping mode) based on PostgreSQL streaming replication. Other software such as cluster software is not required.

This mode replicates the database on all servers that comprise the cluster system. It achieves this by transferring the updated transaction logs of the database from the server that receives the updates (primary server) to another server (standby server), and then reflecting them on the standby server. The client driver automatically distinguishes between the primary and standby servers, so applications can be connected transparently regardless of the physical server.

It consists of a feature that detects faults in the elements that are essential for the continuity of the database operation (such as the database process, disk, and network), as well as simplified switchover and standby server disconnection features. Furthermore, referencing can be performed on the standby server. The database will be copied in synchronous mode.

Note

If using WebAdmin or Mirroring Controller, Fujitsu Enterprise Postgres supports cluster systems comprising one primary server and one standby server.

- Although it is possible to connect an asynchronous standby server to the cluster system as an additional server, the standby server is not targeted for monitoring by Mirroring Controller.
- A synchronous standby server cannot be connected to the cluster system as an additional server.

See

The streaming replication feature is not described in this manual.

Refer to "High Availability, Load Balancing, and Replication" in the PostgreSQL Documentation for information on the streaming replication feature.

Figure 1.1 Failover from the primary server to the standby server

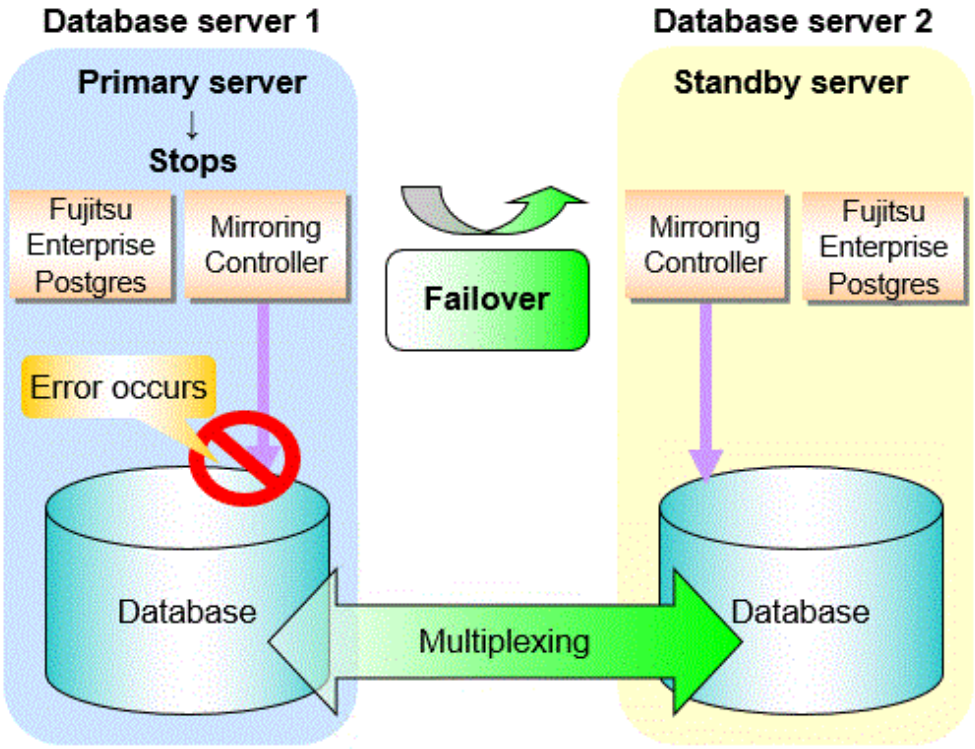
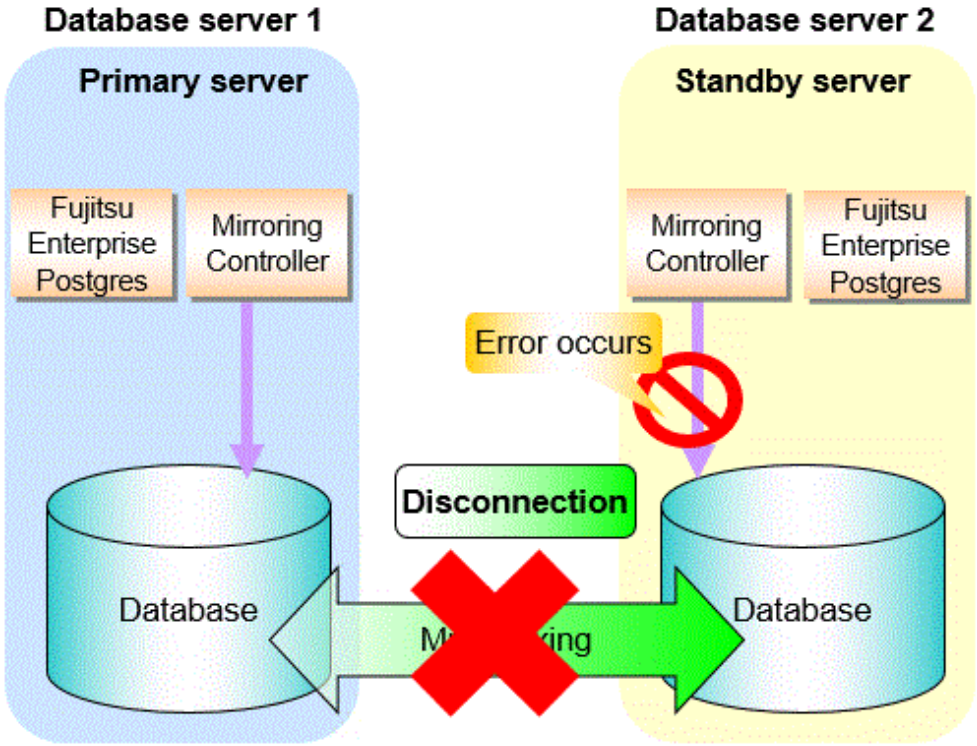


Figure 1.2 Standby server disconnection



Database degradation using the arbitration server

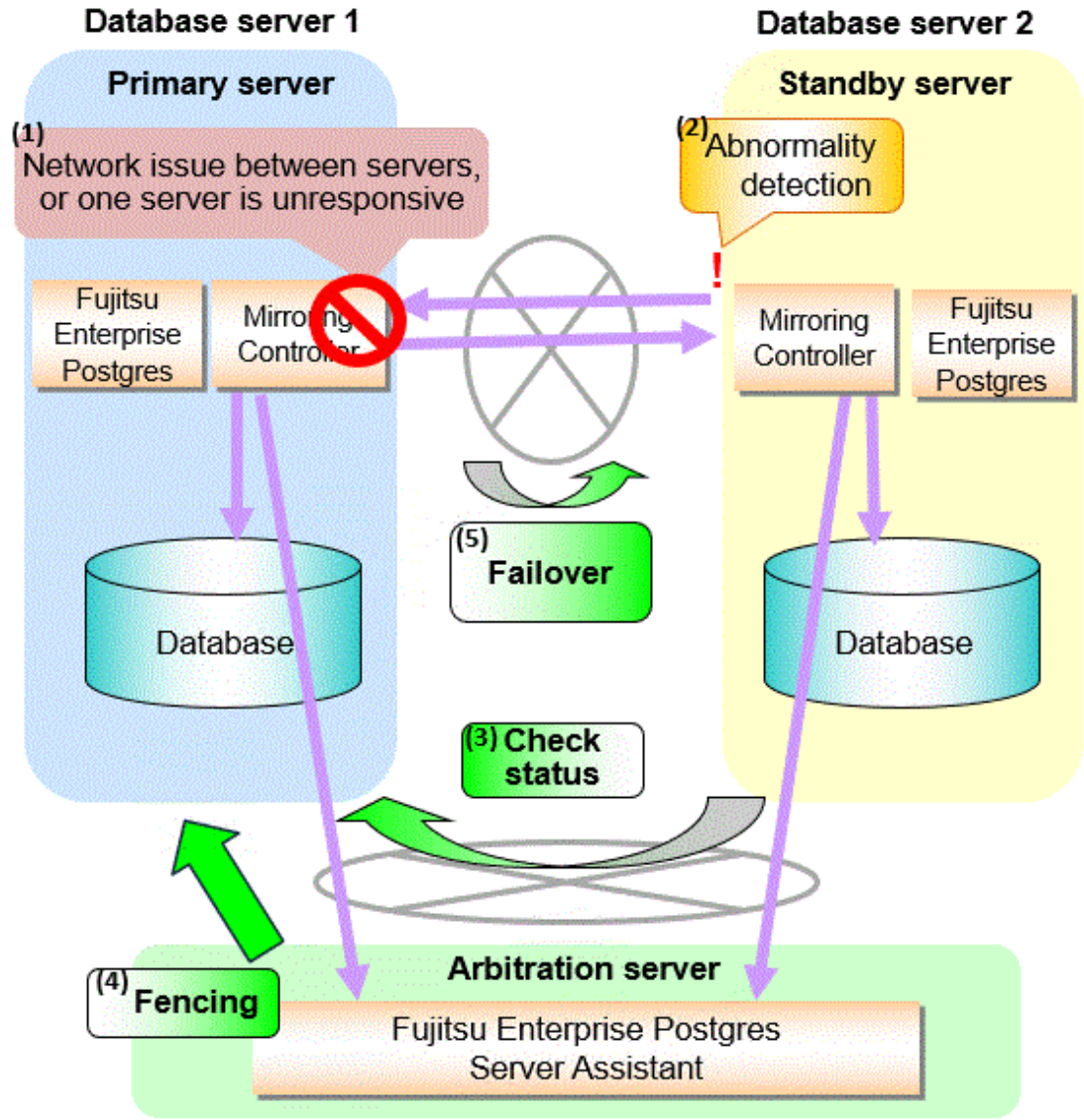
Fujitsu Enterprise Postgres provides the Server Assistant that objectively determines the status of database servers as a third party, and if necessary, isolates affected databases if the database servers are unable to accurately ascertain their mutual statuses in database multiplexing

mode, such as due to a network error between database servers, or server instability. Database degradation can be performed by using the server (arbitration server) on which the Server Assistant is installed.

For database degradation using the arbitration server, if the database servers are unable to check their mutual statuses (due to a network error between database servers or server instability), then the database server queries the arbitration server for the status of the other database server. If it is determined based on the heartbeat result that the status is unstable, the applicable database server will be isolated from the cluster system (fencing). The arbitration server periodically heartbeats the database server so that it can respond immediately to queries from the database server. The fencing process can be customized according to the environment where Mirroring Controller is used.

Additionally, the database servers are always performing their heartbeats for the arbitration server so that it can perform check requests any time.

Figure 1.3 Database degradation using the arbitration server



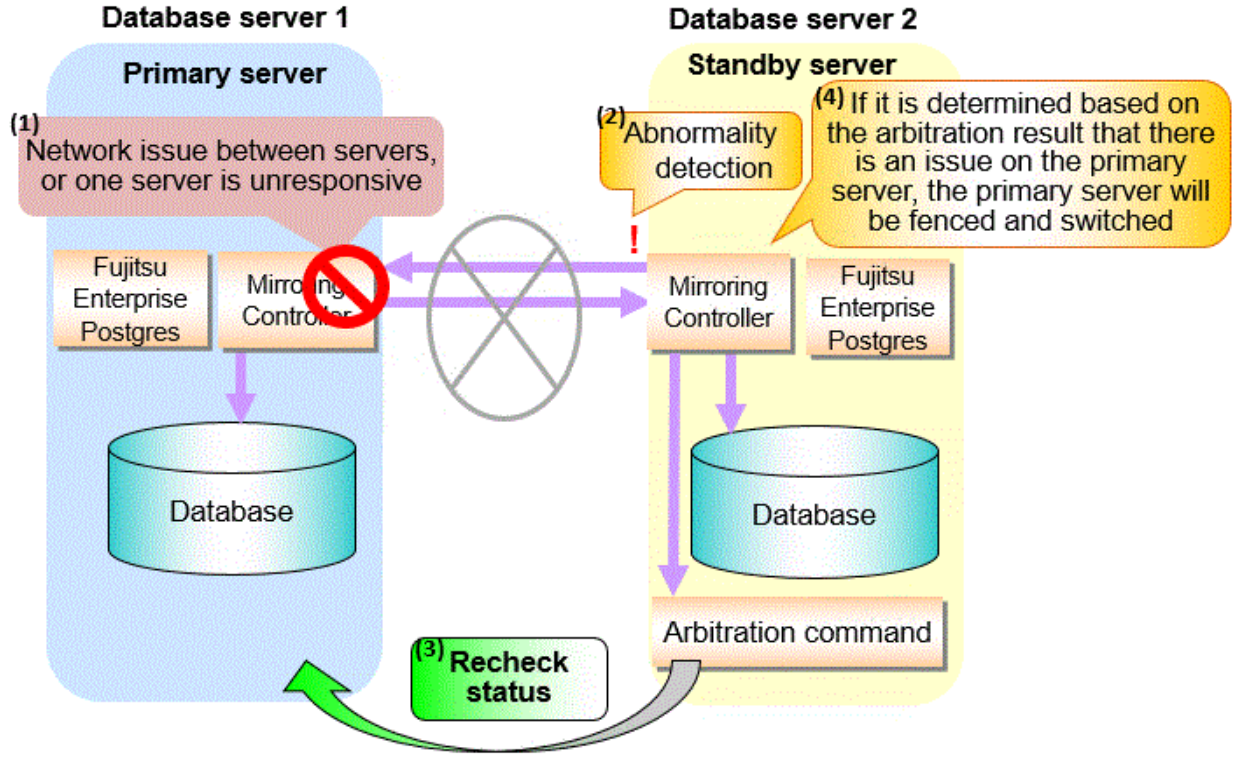
Note

Install the arbitration server on a different physical server to that of the database server. Refer to "1.2 System Configuration for Database Multiplexing Mode" for information on the system configuration when using the arbitration server.

Database degradation using the arbitration command

The arbitration command is a user command that performs arbitration processing in lieu of the arbitration server. If an arbitration server cannot be deployed, arbitration of the database server can be performed using the arbitration command.

Figure 1.4 Database degradation using the arbitration command



See

Refer to "2.6 Creating a User Exit for a Database Server" or "Appendix C User Commands" for information on user commands.

1.1.1 Monitoring Using Database Multiplexing Mode

In database multiplexing mode, perform the monitoring below.

- Operating system or server failures, and no-response state

  By generating a heartbeat between Mirroring Controller on each server, operating system or server errors are detected and acknowledged between the relevant servers.

  The optimal operating method for environments where database multiplexing mode is performed can be selected from the following:

  - Use the arbitration server to perform automatic degradation (switch/disconnect)

    This is the default method.

    The arbitration server objectively determines the status of database servers, then isolates and degrades from the cluster system the ones with an unstable status.

    Refer to "Database degradation using the arbitration server" for details.

  - Call the user exit (user command) that will perform the degradation decision, and perform automatic degradation

    If the arbitration server cannot be installed, select if arbitration processing can be performed by the user instead.

    Mirroring Controller queries the user exit on whether to degrade. The user exit determines the status of the database server, and notifies Mirroring Controller whether to perform degradation.

Refer to "Database degradation using the arbitration command" for details.

- Notification messages

  Use this method if using a two-database server configuration.

  Mirroring Controller outputs messages to the event log when an abnormality is detected. This ensures that a split brain will not occur due to a heartbeat abnormality - however, automatic switching will not be performed if the primary server operating system or server fails or becomes unresponsive.

- Perform automatic degradation unconditionally after a heartbeat abnormality

  This method is handled as in FUJITSU Enterprise Postgres 9.6 or earlier versions.

  This method is not recommended, because Mirroring Controller unconditionally will perform automatic degradation after heartbeat abnormalities.

- Database process failures, and no-response state

  Mirroring Controller periodically accesses the database processes and checks the status. A process error is detected by monitoring whether an access timeout occurs.

- Disk failure

  Mirroring Controller periodically creates files on the data storage destination disk below. A disk error is detected when an I/O error occurs.

  - Data storage destination disk
  - Transaction log storage destination disk
  - Tablespace storage destination disk

  Failures that can be detected are those that physically affect the entire system, such as disk header or device power failures.

- Streaming replication issue

  Mirroring Controller detects streaming replication issues (log transfer network and WAL send/receive processes) by periodically accessing the PostgreSQL system views.

- Mirroring Controller process failure and no response

  In order to continue the monitoring process on Mirroring Controller, Mirroring Controller process failures and no responses are also monitored.

  The Mirroring Controller monitoring process detects Mirroring Controller process failures and no responses by periodically querying the Mirroring Controller process. If an issue is detected, Mirroring Controller is automatically restarted by the Mirroring Controller monitoring process.

Point

- If output of messages is selected as the operation to be performed when a heartbeat abnormality is detected during heartbeat monitoring of the operating system or server, automatic degradation will not be performed.
  However, if an issue in the WAL send process is detected on the primary server, then the standby server will be disconnected, and as a result an automatic disconnection may be performed even if the standby server operating system or server fails or becomes unresponsive.
- You can select in the parameters if the primary server will be switched if a database process is unresponsive or if tablespace storage destination disk failure is detected. Refer to "Appendix A Parameters" for details.
- If the standby server was disconnected, Mirroring Controller will automatically comment out the synchronous_standby_names parameter in the postgresql.conf file of the primary server. Accordingly, you can prevent the application processing for the primary server being stopped.

Note

If the role of primary server was switched to another server and then starts degrading, the original primary server will not become the standby server automatically. Remove the cause of the error, and then change the role of the original primary server to the server currently acting as standby server. Refer to "4.1 Action Required when Server Degradation Occurs" for details.

1.1.2 Referencing on the Standby Server

1.1.2.1 If Prioritizing the Main Job on the Primary Server

If a reference job is performed on the standby server and the primary server is switched, this may impact the main job from the point of view of load and conflict. This is because, on the new primary server (that is, the original standby server), both the main job that was being executed on the original primary server and the reference job that was being continued on the original standby server will be processed.

Therefore, to degrade the reference job (so that the impact on the main job is reduced), you can select the user exit below to disconnect the reference job that was performed on the original standby server.

- Post-switch command

Note

If continuing with the referencing job after switching the primary server, give careful consideration to the server resource estimates, and the likely impact on performance.

1.1.2.2 If Performing the Referencing Job on the Synchronous Standby Server

If an issue such as a log transfer network failure obstructs the continuation of a job on the primary server, the standby server may be automatically disconnected from the cluster system.

Therefore, if operating the reference job on the assumption that the connection destination is the synchronous standby server, you can select to temporarily stop the job by using the user exit or the feature below, so that unexpected referencing of past data does not occur as a result of the disconnection.

- Pre-detach command
- Forced stoppage of the standby server instance on disconnection (specify in the parameter of the server configuration file)

Additionally, if the standby server is incorporated into the cluster system, reference jobs can be started or resumed by using the user exit below.

- Post-attach command

See

- Refer to "2.6 Creating a User Exit for a Database Server" or "Appendix C User Commands" for information on each user exit.
- Refer to "A.4.1 Server Configuration File for the Database Servers" for information on the server configuration file of the database server.

Point

Mirroring Controller will continue processing regardless of the processing result of the above user exits and features.

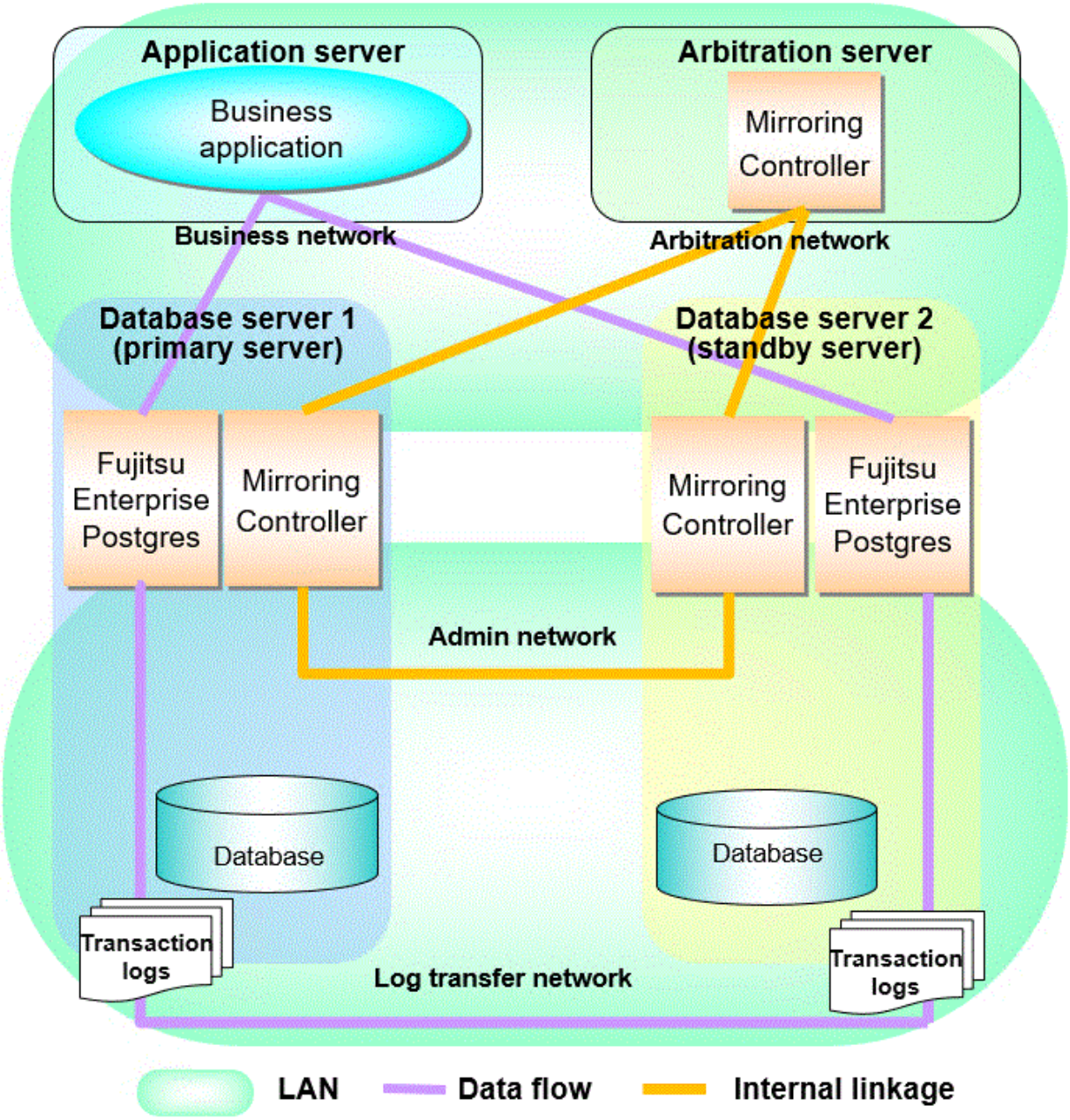
1.2 System Configuration for Database Multiplexing Mode

This section explains the products, features, and networks that are part of a database multiplexing system.

The following table shows the network types uses by database multiplexing systems.

Network type	Description
Job network	Network between the application that accesses the database, and the database server.
Arbitration network	Network used by the arbitration server to check the status of the primary server and standby server, and communicate with Mirroring Controller of the database servers. Additionally, if the job network is disconnected from outside, it can also be used as the arbitration network. Refer to "1.4 Security in Database Multiplexing" for details on network security.
Admin network	Network used by the primary server and the standby server to monitor each other using Mirroring Controller, and to control Mirroring Controller of other servers.
Log transfer network	Network used to transfer the transaction logs of the database, which is part of database multiplexing.

Figure 1.5 System configuration for database multiplexing mode



The arbitration server is installed to check the database server status as a third party, and to perform fencing. Therefore, to obtain the intended benefits, consider the following.

- Install the arbitration server on a different server to that of the database server.
- For the arbitration network, use a network that will not be impacted by line faults or the load on the admin network or log transfer network. This is necessary to correctly determine issues on the admin network or log transfer network.

Point

- The arbitration server can also be used as an application server. However, consider the server load.
- It is recommended to link the arbitration server with other cluster systems, in order to provide redundancy.
- Use the arbitration server in combination with the same version of Fujitsu Enterprise Postgres as that of the primary server and standby server.

- The arbitration server can be built on a different platform to that of the database server.

Note

Because the ping command of the operating system is used for heartbeat monitoring of the database server, configure the network so that ICMP can be used on the admin network and the arbitration network.

1.2.1 Mirroring Controller Resources

This section describes the database server and arbitration server resources of Mirroring Controller.

1.2.1.1 Database Server Resources

The only Mirroring Controller resource is the Mirroring Controller management directory, which stores the files that define the Mirroring Controller behavior, and the temporary files that are created when Mirroring Controller is active.

Note

- Do not create the Mirroring Controller management directory in a directory managed by Fujitsu Enterprise Postgres, otherwise it may be deleted by mistake or may cause unexpected problems when Fujitsu Enterprise Postgres recovery is performed (such as old version of files being restored).

  Refer to "Preparing Directories for Resource Deployment" in the Installation and Setup Guide for Server for information on the directories managed by Fujitsu Enterprise Postgres.

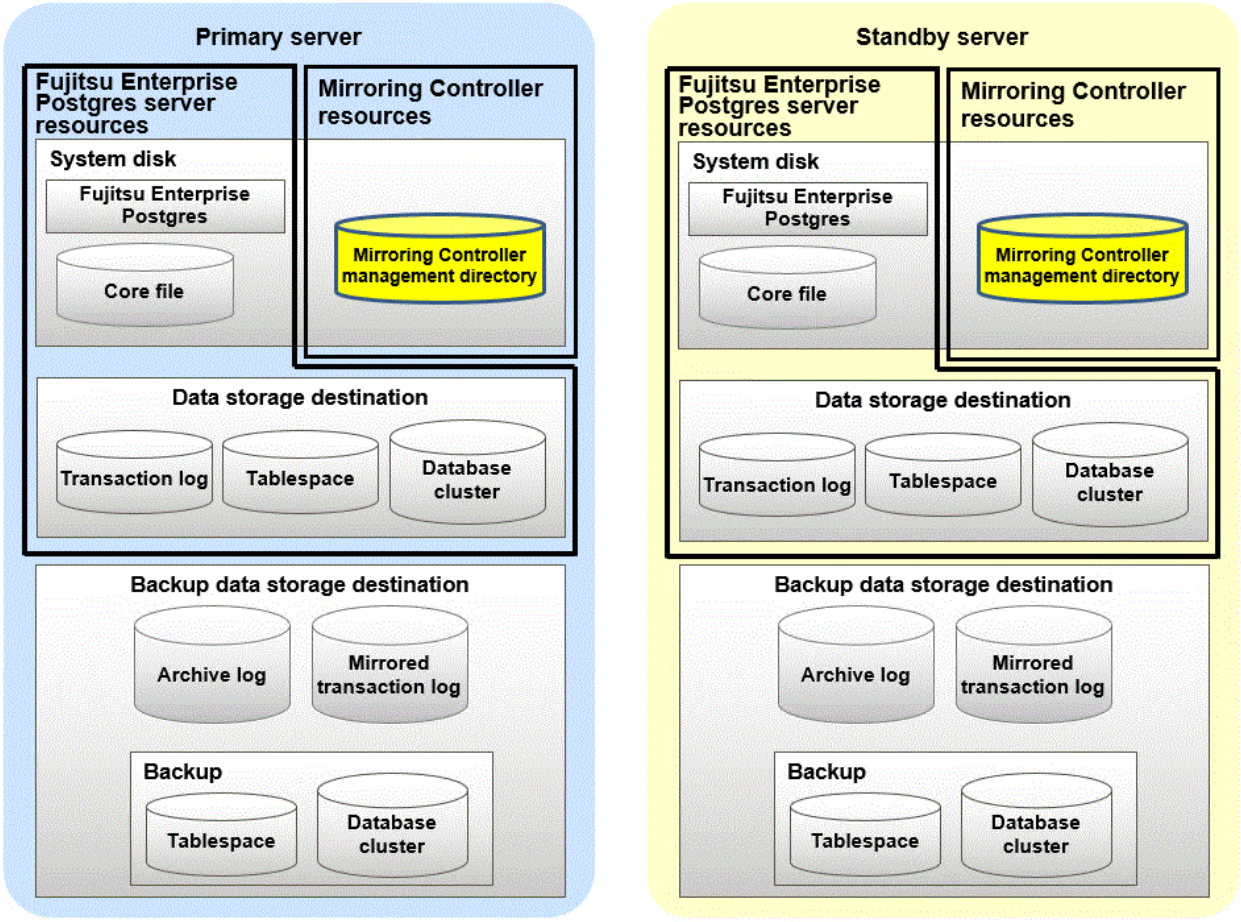
- The backup methods described in "Backing Up the Database" in the Operation Guide cannot be used to back up the Mirroring Controller resources. Therefore, users must obtain their own backup of Mirroring Controller resources, in addition to Fujitsu Enterprise Postgres server resources. Retrieve backups after stopping Mirroring Controller.

- If the automatic switch/disconnection is enabled, do not edit synchronous_standby_names for the Mirroring Controller monitoring target instance. Otherwise, if Mirroring Controller is switched after editing, data may be lost or SQL access may stop.

- If you are building on a virtual machine or cloud, make sure the virtual machines are on different physical servers. Refer to your virtual machine software and cloud vendor documentation for instructions on how to deploy virtual machines.

The content on the primary server will be backed up. You cannot tell which server is the primary server to be backed up, because switching and failback may be performed between the servers. It is also impossible to tell which server is to be restored using the backed up data. Accordingly, ensure that you create a backup of each server when it is working as the primary server.

Figure 1.6 Configuration when backing up Mirroring Controller resources



1.2.1.2 Arbitration Server Resources

The only arbitration server resource is the Mirroring Controller arbitration process management directory. This directory stores the files that define the Mirroring Controller arbitration process behavior and the temporary files created when Mirroring Controller is active.

1.2.2 Mirroring Controller Processes

This section describes the database server and arbitration server processes of Mirroring Controller.

1.2.2.1 Database Server Processes

The database server processes comprise the Mirroring Controller process and Mirroring Controller monitoring process.

Process type	Description
Mirroring Controller process	Performs operating system/server and process heartbeat monitoring and disk abnormality monitoring between database servers. Additionally, it issues arbitration requests to the arbitration server.
Mirroring Controller monitoring process	Performs heartbeat monitoring of the Mirroring Controller process. If the Mirroring Controller process returns no response or is down, the monitoring process is restarted automatically.

1.2.2.2 Arbitration Server Process

The only arbitration process is the Mirroring Controller arbitration process.

Process type	Description
Mirroring Controller arbitration process	Performs rechecks for issues detected on the primary server or the standby server. Additionally, this process performs fencing if it determines that there is an issue on the primary server or the standby server.

1.2.3 Redundancy of the Admin and Log Transfer Networks

The admin network is an important one, because it is used by Mirroring Controller to check the status of each server.

Additionally, the log transfer network is an important one, because it is necessary to ensure data freshness.

Accordingly, configure a failure-resistant network by implementing network redundancy via the NIC teaming feature provided by the operating system or network driver vendor.

Information

NIC teaming feature is provided by the operating system by default.

1.2.4 Notes on CPU Architecture and Products

Use the same CPU architecture (endian) for the primary server, standby server, and the arbitration server.

A server using only PostgreSQL streaming replication cannot be specified as the database multiplexing system log transfer destination.

1.3 Deciding on Operation when a Heartbeat Abnormality is Detected

The operation to be performed when a heartbeat abnormality is detected using operating system/server heartbeat monitoring is decided on according to the environment where database multiplexing mode is performed or the operating method.

It is possible to select from the four operations below, and specify this in the parameters of Mirroring Controller:

- Use the arbitration server to perform automatic degradation (switch/disconnect)
- Call the user exit (user command) that will perform the degradation decision, and perform automatic degradation
- Notification messages
- Perform automatic degradation unconditionally (switch/disconnect)

The table below shows if jobs can be continued on the primary server when an issue is detected during heartbeat monitoring of the operating system/server.

Continuation of jobs on the primary server when an issue is detected during heartbeat monitoring of the operating system/server

Operation	Abnormal event				
	Server/operating system failures or no responses		Admin network issue	Log transfer network issue	Issue on a network for both admin and log transfer
	Primary server	Standby server			
Automatic degradation using the arbitration server	Y (switch)	Y (disconnect)	Y	Y (disconnect)	Y (disconnect)
Call a user exit and perform automatic degradation	Y (switch)	Y (disconnect)	Y	Y (disconnect)	Y (disconnect)

Operation	Abnormal event				
	Server/operating system failures or no responses		Admin network issue	Log transfer network issue	Issue on a network for both admin and log transfer
	Primary server	Standby server			
Notification messages	N (message notification only)	N (message notification only)	Y	Y (disconnect)	Y (disconnect)
Unconditional automatic degradation	Y (switch)	Y (disconnect)	N (split brain occurs)	Y (disconnect)	N (split brain occurs)

Y: Job can be continued

N: Job cannot be continued

1.4 Security in Database Multiplexing

The database server replicates the database on all servers that comprise the cluster system. It achieves this by transferring and reflecting the updated transaction logs of the database from the primary server to the standby server.

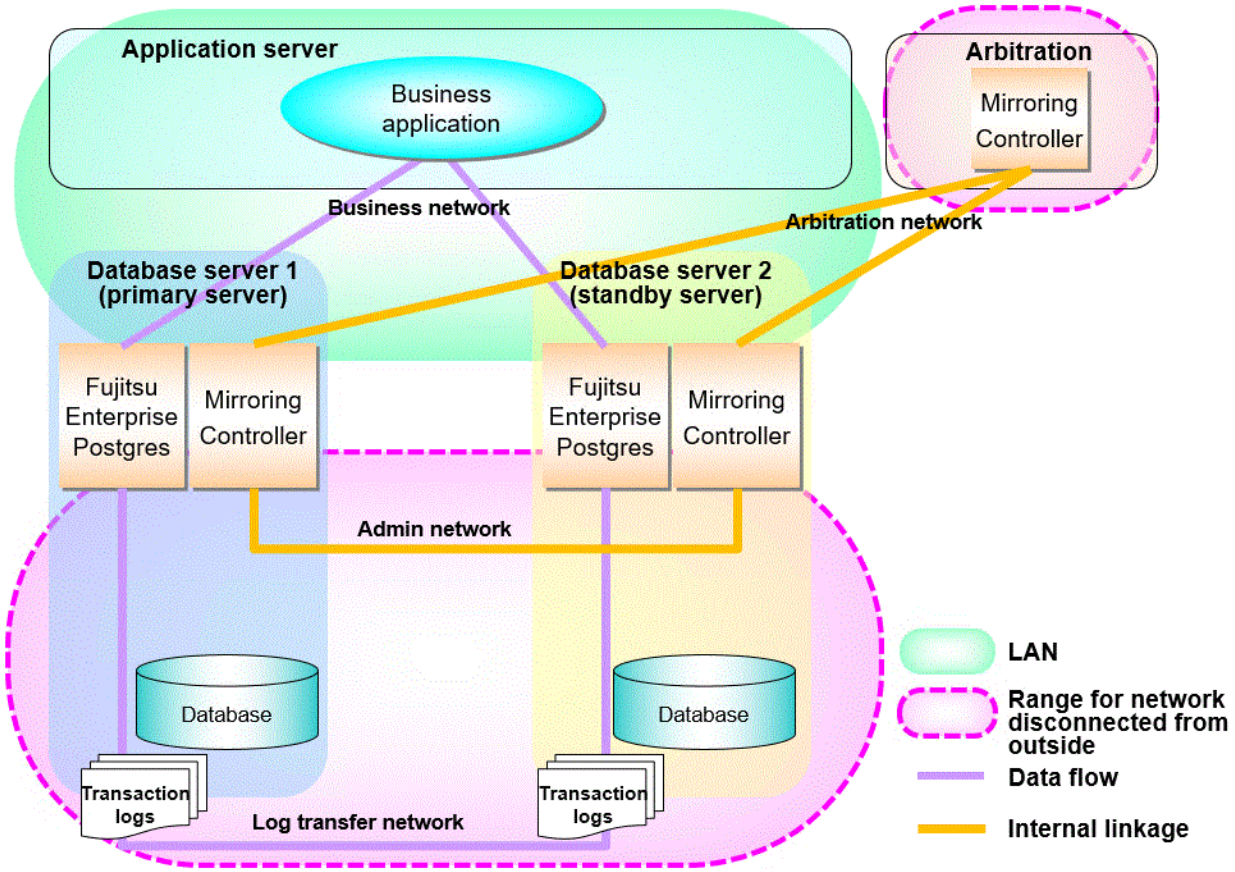
To safeguard the database against unauthorized access and preserve data confidentiality in transaction log transfers, carefully consider security and take note of the following when performing database multiplexing:

- Do not use trust authentication when using replication connection.
- Configure the admin network and the log transfer network so that they cannot be connected from the outside, as shown in Figure 1.7 Security.

Additionally, for the line on which Mirroring Controller connects from the database server to the arbitration server, take note of the following points and consider security carefully.

- Build a network with the arbitration server disconnected from outside, as shown in Figure 1.7 Security.

Figure 1.7 Security



However, it may not always be possible to adopt the configuration mentioned above. For example, you may want to place the servers in a nearby/neighboring office to minimize network delays.

In this case, combine the following features to enhance security:

- Authentication of the Standby Server
- Encryption of Transaction Logs Transferred to the Standby Server

When these features are combined, security will be achieved as shown below.

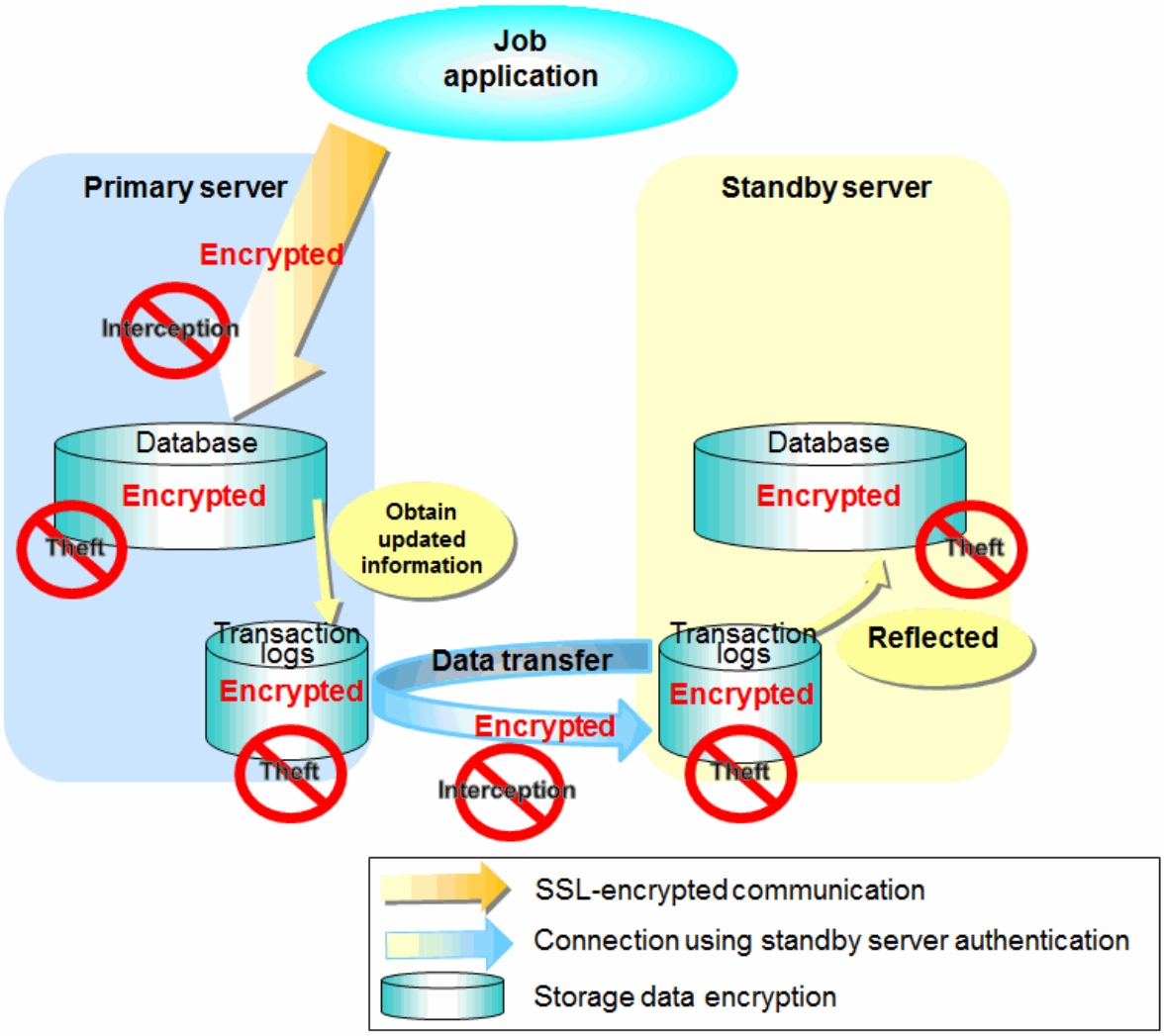
Point

If the job network is disconnected from outside, it can be used as the arbitration network. However, if a network is to be used as both a job network and arbitration network, consider the load on the network.

Note

If a port is blocked (access permission has not been granted) by a firewall, etc., enable use of the target port by granting access. Refer to the vendor document for information on how to open (grant access permission to) a port. Consider the security risks carefully when opening ports.

Figure 1.8 Security achieved when standby server authentication is combined with transaction log encryption



See

Refer to "Performing Database Multiplexing" under "Configuring Secure Communication Using Secure Sockets Layer" in the Operation Guide for information on encrypting SSL communications.

1.4.1 Authentication of the Standby Server

You can prevent spoofing connections from an external server purporting to be the standby server by using authentication with a user name and password.

Configure the setting in the primary server pg_hba.conf file so that authentication is performed for connections from the standby server in the same way as for connections from the client.

See

Refer to "Client Authentication" in the PostgreSQL Documentation for information on content that can be configured in pg_hba.conf.

1.4.2 Encryption of Transaction Logs Transferred to the Standby Server

In case the authentication of the standby server is breached so that a malicious user purporting to be the standby server can spoof data, the transaction log data can be encrypted to prevent it from being deciphered. The transparent data encryption feature is used to encrypt the data.

See

Refer to "Protecting Storage Data Using Transparent Data Encryption" in the Operation Guide for details.

Chapter 2 Setting Up Database Multiplexing Mode

This chapter describes how to set up database multiplexing mode, and how to check it.

Users who perform setup and operations on the database server

Setup and operations of the database server must be performed by the instance administrator user with administrator privileges (operating system user ID belonging to the Administrators group). The instance administrator user must be assigned the "Log on as a service" user right.

Users who perform setup and operations on the arbitration server

The following users may perform setup and operations on the arbitration server when it is used for automatic degradation.

Linux

Any operating system user.

Windows

Any user with administrator privileges. This user must be assigned the "Log on as a service" user right.

Point

- Mirroring Controller selects a database superuser as the user who will connect to the database instance. This enables instance administrator users and database superusers who operate the Mirroring Controller commands to run database multiplexing mode in different environments.
- The application name for connecting to the database instance is "mc_agent".

Matching the system times

Before starting the setup, ensure that the times in the primary server, standby server and arbitration server match, by using the operating system time synchronization feature, for example.

The tolerated difference is approximately one second.

If the system times are not synchronized (because the tolerated difference is exceeded, for example), problem investigation may be affected.

Configuring ICMP

Because the ping command of the operating system is used for heartbeat monitoring of the database server, configure the network so that ICMP can be used on the admin network and the arbitration network. Refer to the relevant operating system procedure for details.

Setup

The setup procedure is shown in the table below. However, the procedure on the arbitration server should be performed only when the arbitration server is used for automatic degradation. A distinction is made between the procedures on the primary server and standby server according to whether the arbitration server is used.

Step	Task			Refer to
	Primary server	Standby server	Arbitration server	
1	Installation			2.1 Installation
2	Preparing the database server		Preparing the arbitration server	2.2 Preparing for Setup
3			Configuring the arbitration server	2.3.1 Configuring the Arbitration Server
4			Creating a user exit	2.3.2 Creating a User Exit for the Arbitration Server

Step	Task			Refer to
	Primary server	Standby server	Arbitration server	
5			Starting the arbitration process	2.3.3 Starting the Mirroring Controller Arbitration Process
6	Setting up database multiplexing mode			2.4.1 Setting Up Database Multiplexing Mode on the Primary Server
7	Creating, setting, and registering the instance			2.4.2 Creating, Setting, and Registering the Primary Server Instance
8	Creating a user exit			2.6 Creating a User Exit for a Database Server
9	Starting Mirroring Controller			2.4.3 Starting Mirroring Controller on the Primary Server
10		Setting up database multiplexing mode		2.5.1 Setting Up Database Multiplexing Mode on the Standby Server
11		Creating, setting, and registering the instance		2.5.2 Creating, Setting, and Registering the Standby Server Instance
12		Creating a user exit		2.6 Creating a User Exit for a Database Server
13		Starting Mirroring Controller		2.5.3 Starting Mirroring Controller on the Standby Server
14	Confirming the streaming replication status			2.7 Confirming the Streaming Replication Status
15	Checking the connection status			2.8.1 Checking the Connection Status on a Database Server
16		Checking the connection status		2.8.1 Checking the Connection Status on a Database Server
17			Checking the connection status	2.8.2 Checking the Connection Status on the Arbitration Server
18	Creating applications			2.9 Creating Applications
19	Checking the behavior			2.10 Checking the Behavior

Explanations for each step are provided below.

Information

- The setup procedure is also the same when changing the mode on a single server to database multiplexing mode. In this case, omit the installation of Fujitsu Enterprise Postgres and the creation of the instance.

  Refer to "3.9.2 Changing from Single Server Mode to Database Multiplexing Mode" for details.

- The primary and standby server can be pseudo-configured on the same server for system testing, for example. In this case, the setup can be performed using the same procedure, however there will be some supplementary steps.

  Before performing the setup, refer to "Appendix B Supplementary Information on Building the Primary Server and Standby Server on the Same Server".

2.1 Installation

Refer to the manuals below, and then install the product.

See

- Refer to the Installation and Setup Guide for Server for details on how to install Fujitsu Enterprise Postgres.
- Refer to the Installation and Setup Guide for Server Assistant for information on installing the Server Assistant on the arbitration server.

Note

Do not use the arbitration server also as a database server. The arbitration server is installed to check the database server status as a third party, and to perform fencing. Using the arbitration server also as a database server nullifies the effectiveness of the arbitration server.

2.2 Preparing for Setup

This section describes the preparation required before setting up Mirroring Controller.

2.2.1 Preparing the Database Server

2.2.1.1 Preparing the Backup Disk

In Mirroring Controller, by performing a backup, recovery is possible even if all server disks are corrupted.

The content on the primary server should be backed up. However, through switching and failback, the standby server may also become the primary server. Accordingly, prepare each of the backup disk devices for the primary and standby servers. Perform backup on the primary server used at the time of the backup.

2.2.1.2 Preparatory Tasks for the Output of Error Logs to the Event Log

This section explains the preparatory tasks required to output error logs to the event log.

Note

If you do not register an event source name, the message content output to the event log may be incomplete.

Setting each server

You should register this default event source name beforehand because the default event source name "MirroringControllerOpen" may be output to the event log when Mirroring Controller commands on the database server are used.

Example)
The following is an example in which the DLL of a 64-bit product is registered under the default event source name. Note that "<*x*>" indicates the product version.

```
> regsvr32 "c:\Program Files\Fujitsu\fsepv<x>server64\lib\mcevent.dll"
```

Setting each instance

You can output messages to any event source named by the user, so that messages output to the event log can be identified by each instance.

Example)
The following is an example in which the DLL of a 64-bit product is registered under the event source name "Mirroring Controller inst1". Note that "<*x*>" indicates the product version.

```
> regsvr32 /n /i:"Mirroring Controller inst1" "c:\Program Files\Fujitsu\fsepv<x>server64\lib
\mcevent.dll"
```

The parameter must be edited for each instance. Refer to "A.4.1 Server Configuration File for the Database Servers" to set the event_source parameter.

If installing multiple versions

If Fujitsu Enterprise Postgres is already installed on the same machine, search for the key below in a registry editor, and make a note of the path of the registered DLL. Afterwards, register a new DLL using the default event source name.

Use the DLL path that you made a note of in the above step when re-registering the default event source name during uninstallation.

```
MirroringControllerOpen
```

2.2.1.3 Security Policy Settings

Security settings that allow logon as a service are required in Mirroring Controller for the operating system user account of the instance administrator user in order to start and stop Mirroring Controller and an instance using a Windows service.

If the security settings have not been configured, refer to "E.1 Security Policy Settings" for information on configuring the settings.

2.2.2 Preparing the Arbitration Server

2.2.2.1 Preparing to Output Error Logs to the Event Log (Windows)

This section explains the preparatory tasks for outputting error logs to the event log.

If no event source name is registered, messages output to the event log may be incomplete.

Configuring each server

Event logs for the Mirroring Controller commands on the arbitration server may be output with the default event source name "MirroringControllerArbiter". Therefore, register this default event source name beforehand.

Example)
The following is an example in which the DLL of a 64-bit product is registered under the default event source name. Note that "<*x*>" indicates the product version.

```
> regsvr32 "c:\Program Files\Fujitsu\fsepv<x>assistant64\lib\mcarbevent.dll"
```

Setting each instance

You can output messages to any event source named by the user, so that messages output to the event log can be identified by each instance.

Example)
The following is an example in which the DLL of a 64-bit product is registered under the event source name "Mirroring Controller arbiter1". Note that "<*x*>" indicates the product version.

```
> regsvr32 /n /i:"Mirroring Controller arbiter1" "c:\Program Files\Fujitsu\fsepv<x>assistant64\lib
\mcarbevent.dll"
```

The parameter must be edited for each instance. Refer to "A.4.2 Arbitration Configuration File" and set the event_source parameter.

If installing multiple versions

If Fujitsu Enterprise Postgres is already installed on the same machine, search for the key below in Registry Editor, and make a note of the path of the registered DLL. Afterwards, register a new DLL using the default event source name.

Use the DLL path that you made a note of in the above step when re-registering the default event source name during uninstallation.

```
MirroringControllerArbiter
```

2.2.2.2 Security Policy Settings (Windows)

On the arbitration server, operating system user accounts that operate the Mirroring Controller arbitration process must be assigned the "Log on as a service" user right in order to use Windows Services to start and stop the Mirroring Controller arbitration process.

If the security settings to enable this have not been configured, refer to "E.1 Security Policy Settings" and configure the settings.

2.3 Setting Up the Arbitration Server

This section explains how to set up the arbitration server.

2.3.1 Configuring the Arbitration Server

This section explains how to set up database multiplexing mode on the arbitration server.

In database multiplexing mode, the files that are required for operations are managed in the Mirroring Controller arbitration process management directory.

There is one Mirroring Controller arbitration process management directory for each arbitration process.

Point

The arbitration process for each database multiplexing system can be started on a single arbitration server.

See

- Refer to the Reference for information on the mc_arb command.
- Refer to "Appendix A Parameters" for information on the parameters to be edited for the setup.

Perform the following procedure:

Linux

1. On the arbitration server, log in as any operating system user who starts and stops the arbitration process.
2. Configure the environment variables.

   Set the following environment variables:

   - PATH

     Add the installation directory "/bin".

   - MANPATH

     Add the installation directory "/share/man".

   Example

   The following example configures environment variables when the installation directory is "/opt/fsepv<*x*>assistant".

   Note that "<*x*>" indicates the product version.

   sh, bash

   ```
   $ PATH=/opt/fsepv<x>assistant/bin:$PATH ; export PATH
   $ MANPATH=/opt/fsepv<x>assistant/share/man:$MANPATH ; export MANPATH
   ```

   csh, tcsh

   ```
   $ setenv PATH /opt/fsepv<x>assistant/bin:$PATH
   $ setenv MANPATH /opt/fsepv<x>assistant/share/man:$MANPATH
   ```

3. Create the Mirroring Controller arbitration process management directory that will store the files required by the arbitration server.

   Use ASCII characters in the Mirroring Controller arbitration process management directory.

4. In the network configuration file (network.conf), define the Mirroring Controller network configuration that will be managed by the Mirroring Controller arbitration process.

   Create network.conf in the Mirroring Controller arbitration process management directory, based on the sample file. For network.conf, set read and write permissions only for the operating system user who starts and stops the arbitration process in step 1.

   If users other than this are granted access permissions, the mc_arb command will not work. Accordingly, users other than the operating system user who starts and stops the arbitration process in step 1 are prevented from operating the Mirroring Controller arbitration process.

   Sample file

   ```
   /installDir/share/mcarb_network.conf.sample
   ```

   In network.conf, specify the IP address or host name and port number of the primary server and standby server, and define the Mirroring Controller network configuration that will be managed by the Mirroring Controller arbitration process.

   Refer to "A.3 Network Configuration File" for details.

   A definition example is shown below.

   ```
   Example)
   The IDs of the servers are set to "server1" and "server2", and their port numbers are set to
   "27541".
   ```

   ```
   server1 192.0.3.100 27541
   server2 192.0.3.110 27541
   ```

5. In the arbitration configuration file (arbitration.conf), define the information related to control of the Mirroring Controller arbitration process.

   Create arbitration.conf in the Mirroring Controller arbitration process management directory, based on the sample file. For arbitration.conf, set read and write permissions only for the operating system user who starts and stops the arbitration process in step 1. If users other than this are granted access permissions, the mc_arb command will not work.

   Sample file

   ```
   /installDir/share/mcarb_arbitration.conf.sample
   ```

   Set the parameters shown in the table below in arbitration.conf.

Table 2.1 Parameters

Parameter	Content specified	Remarks
port	Port number of the Mirroring Controller arbitration process	The port number must be 0 to 65535. Ensure that the port number does not conflict with other software. Do not specify an ephemeral port that may temporarily be assigned by another program.
my_address	'*ipAddrOrHostNameThatAcceptsConnectionFromMirroringControllerProcessOnDbServer*' [Setting example] my_address = '192.0.3.120'	IPv4 and IPv6 addresses can be specified. Specify the IP address, enclosed in single quotation marks (').
syslog_ident	'*programName*'	Specify using single quotation marks (') to enclose the program name used to identify the Mirroring Controller arbitration process message in the system log. Use ASCII

Parameter	Content specified	Remarks
		characters excluding spaces to specify this parameter. The default is 'MirroringControllerArbiter'.
fencing_command	'*fencingCmdFilePath*' [Setting example] fencing_command = '/arbiter/fencing_dir/execute_fencing.sh'	Specify the full path of the fencing command that fences a database server where it is determined that an error has occurred. Enclose the path in single quotation marks ('). Specify the path using less than 1024 bytes.
fencing_command_timeout	Timeout for fencing command (seconds)	If the command does not respond within the specified number of seconds, it is determined that fencing has failed and a signal (SIGTERM) is sent to the fencing command execution process. Specify a value between 1 and 2147483647. The default is 20 seconds.

Information

Refer to "A.4.2 Arbitration Configuration File" for information on the parameters and for other parameters.

Windows

1. On the arbitration server, log in as any operating system user who starts and stops the arbitration process.
2. Configure the environment variables.

   Set the following environment variable:

   - PATH

     Add the installation folders "\bin" and "\lib".

   Example

   The following example configures environment variables when the installation folder is "c:\Program Files\Fujitsu \fsepv<*x*>assistant64".

   Note that "<*x*>" indicates the product version.

   ```
   > SET PATH=c:\Program Files\Fujitsu\fsepv<x>assistant64\bin;c:\Program Files\Fujitsu
   \fsepv<x>assistant64\lib;%PATH%
   ```

3. Create the Mirroring Controller arbitration process management directory that will store the files required by the arbitration server.

   Use ASCII characters in the Mirroring Controller arbitration process management directory.

4. In the network configuration file (network.conf), define the Mirroring Controller network configuration that will be managed by the Mirroring Controller arbitration process.

   Create network.conf in the Mirroring Controller arbitration process management directory, based on the sample file.

   Sample file

   ```
   installDir\share\mcarb_network.conf.sample
   ```

   In network.conf, specify the IP address or host name and port number of the primary server and standby server, and define the Mirroring Controller network configuration that will be managed by the Mirroring Controller arbitration process.

Refer to "A.3 Network Configuration File" for details.

A definition example is shown below.

```
Example)
The IDs of the servers are set to "server1" and "server2", and their port numbers are set to
"27541".
```

```
server1 192.0.3.100 27541
server2 192.0.3.110 27541
```

5. Change the access permissions for network.conf.

   For network.conf, set read and write permissions only for the operating system user who starts and stops the arbitration process in step 1. If users other than this are granted access permissions, the mc_arb command will not work. Accordingly, users other than the operating system user who starts and stops the arbitration process in step 1 are prevented from operating the Mirroring Controller arbitration process.

   ```
   Example)
   The following is an execution example, in which the operating system user who starts and stops the
   arbitration process in step 1 is granted full access permissions as the owner when the user is
   "fsepuser". The following procedure applies when the user is logged in to the Windows server as
   "fsepuser".
   ```

   ```
   > takeown /f network.conf
   > icacls network.conf /reset
   > icacls network.conf /inheritance:r
   > icacls network.conf /grant fsepuser:F
   ```

6. In the arbitration configuration file (arbitration.conf), define the information related to control of the Mirroring Controller arbitration process.

   Create arbitration.conf in the Mirroring Controller arbitration process management directory, based on the sample file.

   ```
   Sample file
   ```

   ```
   installDir\share\mcarb_arbitration.conf.sample
   ```

   Set the parameters shown in the table below in arbitration.conf.

   Table 2.2 Parameters

Parameter	Content specified	Remarks
port	Port number of the Mirroring Controller arbitration process	The port number must be 0 to 65535. Ensure that the port number does not conflict with other software. Do not specify an ephemeral port that may temporarily be assigned by another program.
my_address	'*ipAddrOrHostNameThatAcceptsConnectionFromMirroringControllerProcessOnDbServer*' [Setting example] my_address = '192.0.3.120'	IPv4 and IPv6 addresses can be specified. Enclose the parameter value in single quotation marks (').
service_name	'*registeredServiceNameOfArbitrationProcessInWindowsServices*'	Use ASCII characters excluding forward slash (/) and backslash (\) to specify this parameter. Enclose the parameter value in single quotation marks ('). The maximum length of the service name is 124 bytes.
event_source	'*eventSourceNameUsedToIdentifyArbitrationProcessMsgInEventLog*'	Use ASCII characters excluding spaces to specify this parameter.

Parameter	Content specified	Remarks
		Enclose the parameter value in single quotation marks ('). The maximum length of the event source name is 255 bytes.
fencing_command	'*fencingCmdFilePath*' [Setting example] fencing_command = 'c:\\arbiter\\fencing_dir\\execute_fencing.bat'	Specify the full path of the fencing command that fences a database server where it is determined that an error has occurred. Specify "\\" as the delimiter. Enclose the path in single quotation marks ('). Specify the path using less than 260 bytes. Any multibyte characters must use the same encoding as the operating system.
fencing_command_timeout	Timeout for fencing command (seconds)	If the command does not respond within the specified number of seconds, it is determined that fencing has failed and a signal (SIGTERM) is sent to the fencing command execution process. Specify a value between 1 and 2147483647. The default is 20 seconds.

Information

Refer to "A.4.2 Arbitration Configuration File" for information on the parameters and for other parameters.

7. Change the access permissions for arbitration.conf.

   For arbitration.conf, set read and write permissions only for the operating system user who starts and stops the arbitration process in step 1. If users other than this are granted access permissions, the mc_arb command will not work.

   ```
   Example)
   The following is an execution example, in which the operating system user who starts and stops the
   arbitration process in step 1 is granted full access permissions as the owner when the user is
   "fsepuser". The following procedure applies when the user is logged in to the Windows server as
   "fsepuser".
   ```

   ```
   > takeown /f arbitration.conf
   > icacls arbitration.conf /reset
   > icacls arbitration.conf /inheritance:r
   > icacls arbitration.conf /grant fsepuser:F
   ```

8. Configure Windows Firewall.

   If Windows Firewall is used, enable the port number of Mirroring Controller specified in the network configuration file in step 3. Refer to "E.2 Windows Firewall Settings" for details.

9. Register the Mirroring Controller arbitration process as a Windows service.

   Execute the mc_arb command in register mode.

   For the -P option of the mc_arb command, specify the password of the operating system user who executes the command.

   ```
   Example)
   ```

   ```
   > mc_arb register -M D:\mcarb_dir\arbiter1 -P ********
   ```

Note

When specifying the password in the -P option of the mc_arb command, for security reasons, you should be careful not to allow other users to access it.

Information

Use the mc_arb command with the -S option to specify automatic start and stop of the Mirroring Controller arbitration process. Refer to "2.13 Setting Automatic Start and Stop of the Mirroring Controller Arbitration Process" for details.

The Mirroring Controller arbitration process is registered as a Windows service using the service name specified in the service_name parameter of arbitration.conf in step 6.

You can execute the sc qc command to check the registration status.

2.3.2 Creating a User Exit for the Arbitration Server

The only user exit for the arbitration server is the fencing command.

The fencing command is a user command that is called by the Mirroring Controller arbitration process if Mirroring Controller performs arbitration processing and determines that a database server is unstable.

In the fencing command, the user implements a process that isolates a database server from a cluster system by, for example, stopping the target operating system or server. The fencing command that was created is to be specified for the parameter in the arbitration configuration file. Refer to "A.4.2 Arbitration Configuration File" for information on the parameters.

- Fencing the primary server during the switch
  - Prevent the Mirroring Controller management process on the primary server from communicating with the Mirroring Controller management process on the other server.
  - Prevent applications from connecting to the primary server instance.
- Fencing the standby server during disconnection
  - Prevent the Mirroring Controller management process on the standby server from communicating with the Mirroring Controller management process on the other server.
  - Prevent applications from connecting to the standby server instance.
  - Prevent the standby server from continuing streaming replication.

See

Refer to "Appendix C User Commands" for information on user exits.

2.3.3 Starting the Mirroring Controller Arbitration Process

This section explains how to start the Mirroring Controller arbitration process.

An operating system user who has logged in to the arbitration server can start the Mirroring Controller arbitration process by executing the mc_arb command in start mode.

Linux

```
Example)
```

```
$ mc_arb start -M /mcarb_dir/arbiter1
```

Windows

Example)

```
> mc_arb start -M D:\mcarb_dir\arbiter1
```

2.4 Setting Up the Primary Server

This section explains how to set up the primary server.

2.4.1 Setting Up Database Multiplexing Mode on the Primary Server

This section explains how to set up database multiplexing mode on the primary server.

In database multiplexing, the files that are required for operations are managed in the Mirroring Controller management directory.

There is one Mirroring Controller management directory for each instance.

Note

- Do not place the Mirroring Controller management directory in a directory managed by Fujitsu Enterprise Postgres, otherwise it may be deleted by mistake with the directories managed by Fujitsu Enterprise Postgres, and an old version of files may be restored.

See

- Refer to "Preparing Directories for Resource Deployment" in the Installation and Setup Guide for Server for details on the directories that are managed by Fujitsu Enterprise Postgres.
- Refer to "mc_ctl" in Reference for information on the command.
- Refer to "Appendix A Parameters" for details on each parameter to be edited for the setup.

Perform the following procedure:

1. Log in to the primary server.
2. Create the Mirroring Controller management directory that will store the files required by database multiplexing.

   Use ASCII characters in the Mirroring Controller management directory.

   Additionally, grant "Write" permission to the instance administrator user for the Mirroring Controller management directory.
3. In the network configuration file (network.conf), define the network configuration that will link between the Mirroring Controller processes.

   Create the network.conf file in the Mirroring Controller management directory, based on the sample file.

   Sample file

   ```
   installDir\share\mc_network.conf.sample
   ```

   In network.conf, specify the IP address or host name and port number of the primary server and standby server, and define the network configuration that will link between the Mirroring Controller processes, and between Mirroring Controller processes and the Mirroring Controller arbitration process.

   Refer to "A.3 Network Configuration File" for details.

   A definition example is shown below.

   The content to be defined depends on the operation settings at the time a heartbeat abnormality is detected.

When automatic degradation by the arbitration server is selected

```
Example)
The IDs of the primary server and standby server are set to "server1" and "server2", and their
port numbers are set to "27540" and "27541". The ID of the server of the Mirroring Controller
arbitration process is set to "arbiter", and its port number is set to "27541".
```

```
server1 192.0.2.100,192.0.3.100 27540,27541 server
server2 192.0.2.110,192.0.3.110 27540,27541 server
arbiter 192.0.3.120 27541 arbiter
```

Ensure that the port numbers set for the primary server, standby server, and arbitration server do not conflict with other software. In addition, when the arbitration server is used for automatic degradation, use a network in which the arbitration network is not affected by a line failure in the admin network.

When the server type is "server", two IP addresses or host names, and two port numbers need to be specified in the following order:

- IP address or host name of the database server used as the admin network
- IP address or host name of the database server used as the arbitration network
- Port number of the database server used as the admin network
- Port number of the database server used as the arbitration network

If the server type is "arbiter", specify the IP address or host name set for the my_address parameter and the port number set for the port parameter in arbitration.conf of the arbitration server.

When operation other than automatic degradation by the arbitration server is selected

```
Example)
The IDs of the servers are set to "server1" and "server2", and their port numbers are set to
"27540".
```

```
server1 192.0.2.100 27540
server2 192.0.2.110 27540
```

Ensure that the port numbers for the primary and standby server do not conflict with other software.

Register the port number of the primary server in the services file, because there are programs, such as WebAdmin, that search an available port number using the services file.

Register any name as the service name.

4. Change the access permissions for the network.conf file.

   For network.conf, set read and write permissions for the instance administrator user only.

   If users other than the instance administrator user are granted access permissions, the mc_ctl command will not work. Accordingly, users other than the instance administrator user are prevented from operating Mirroring Controller.

   ```
   Example)
   The following is an execution example, in which the instance administrator user is granted full
   access permissions as the owner when the operating system user name of the instance administrator
   user is "fsepuser". The following procedure applies when the user is logged in to the Windows
   server as "fsepuser":
   ```

   ```
   > takeown /f network.conf
   > icacls network.conf /reset
   > icacls network.conf /inheritance:r
   > icacls network.conf /grant fsepuser:F
   ```

5. Define the information related to Mirroring Controller monitoring and control in the *serverIdentifier*.conf file.

   Create the *serverIdentifier*.conf file in the Mirroring Controller management directory, based on the sample file.

   As the file name for the *serverIdentifier*.conf file, use the server identifier name that was specified in the network.conf file in step 3.

   ```
   Sample file
   ```

```
installDir\share\mc_server.conf.sample
```

Set the parameters shown in the table below in the *serverIdentifier*.conf file.

Table 2.3 Parameters

Parameter	Content specified	Remarks
db_instance	'*dataStorageDestinationDir*'	Use ASCII characters, and specify "\\" as the path delimiter. Enclose the parameter value in single quotation marks (').
db_instance_service_name	'*registeredServiceNameOfFujitsuEnterprisePostgresInstance*'	Specify the registered service name of the Fujitsu Enterprise Postgres instance in the Windows service. Use ASCII characters, enclosed in single quotation marks (').
db_instance_password	'*passwordOfInstanceAdminUser*'	If password authentication is performed, specify this parameter in the settings used when Mirroring Controller connects to a database instance. Use ASCII characters, enclosed in single quotation marks ('). If the specified value of this parameter includes ' or \, write \' or \\, respectively.
enable_hash_in_password	on or off	Specify on to treat the # in the db_instance_password specification as a password character, or off to treat it as a comment. The default is "off".
mc_service_name	'*registeredServiceNameOfMirroringController*'	Specify the Mirroring Controller service name registered in the Windows service. Use ASCII characters excluding forward slash (/) and backslash (\) to specify this parameter. The service name is up to 124 bytes.
event_source	'*eventSourceName*'	Specify the event source name to be used to identify the Mirroring Controller message in the event log. Use ASCII characters to specify this parameter. The maximum length of the event source name is 255 bytes. By using a similar event source name as the postgresql.conf file parameter, the Mirroring Controller output content can be referenced transparently, so log reference is easy.
remote_call_timeout	Admin communication timeout	Specify the timeout value (milliseconds) of the Mirroring Controller agent process for communication between servers. Specify a value that is less than the operation system TCP connection timeout. Also, when using the Mirroring Controller arbitrage process for arbitrage, fencing, and state transition commands, specify a value

Parameter	Content specified	Remarks
		that is greater than the sum of the timeout values.
heartbeat_error_action	Operation when a heartbeat abnormality is detected using operating system or server heartbeat monitoring	arbitration: Perform automatic degradation using the arbitration server. command: Call a user exit to determine degradation, and perform automatic degradation if required. message: Notify messages. fallback: Perform automatic degradation unconditionally. Set the same value on the primary server and standby server.
heartbeat_interval	Interval time for abnormality monitoring during heartbeat monitoring of the operating system or server (milliseconds)	Abnormality monitoring of the operating system or server is performed at the interval (milliseconds) specified in heartbeat_interval. This parameter setting is used as the default for database process heartbeat monitoring, streaming replication abnormality monitoring, and disk abnormality monitoring. When setting the monitoring time, there are some considerations to take into account to optimize degradation using abnormality monitoring. Refer to "2.11.4.1 Tuning for Abnormality Monitoring of the Operating System or Server" for details.
heartbeat_timeout	Timeout for abnormality monitoring during heartbeat monitoring of the operating system or server (seconds)	
heartbeat_retry	Number of retries for abnormality monitoring during heartbeat monitoring of the operating system or server (number of times)	
fencing_command	'*fencingCmdFilePath*' [Setting example] fencing_command = 'c:\\mc\\fencing_dir\\execute_fencing.bat'	Specify the full path of the fencing command that fences a database server where an error is determined to have occurred. Enclose the path in single quotation marks ('). Any multibyte characters must use the same encoding as the operating system. This parameter must be specified when "command" is set for heartbeat_error_action. Specify the path using less than 260 bytes.
fencing_command_timeout	Fencing command timeout (seconds)	If the command does not respond within the specified number of seconds, fencing is determined to have failed and a signal (SIGTERM) is sent to the fencing command execution process. Specify a value between 1 and 2147483647. The default is 20 seconds.
arbitration_timeout	Timeout for arbitration processing in the Mirroring Controller arbitration process (seconds)	The specified value must be at least equal to the heartbeat monitoring time of the operating system or server + fencing_command_timeout in the arbitration configuration file.

Parameter	Content specified	Remarks
		If there is no response for at least the number of seconds specified, the primary server will not be switched and the standby server will not be disconnected. Therefore, perform degradation manually. Specify a value between 1 and 2147483647. This parameter does not need to be set for operation that does not use the arbitration server.
arbitration_command	'*ArbitrationCmdFilePath*' [Setting example] arbitration_command = 'c:\\mc\\arbitration_dir\\execute_arbitration_command.bat'	Specify the full path of the arbitration command to be executed when an abnormality is detected during heartbeat monitoring of the operating system or server. Enclose the path in single quotation marks ('). Any multibyte characters must use the same encoding as the operating system. This parameter must be specified when "command" is set for heartbeat_error_action. Specify the path using less than 260 bytes.
arbitration_command_timeout	Timeout for arbitration command (seconds)	If the command does not respond within the specified number of seconds, it is determined that execution of the arbitration command has failed and a signal (SIGTERM) is sent to the arbitration command execution process. Specify a value between 1 and 2147483647. This parameter can be specified only when "command" is set for heartbeat_error_action.

Information

Refer to "A.4.1 Server Configuration File for the Database Servers" for information on the parameters and for other parameters.

6. Change the access permissions for the *serverIdentifier*.conf file.

   For *serverIdentifier*.conf, set read and write permissions for the instance administrator user only. If users other than the instance administrator user are granted access permissions, the mc_ctl command will not work.

   ```
   Example)
   The following is an execution example, in which the instance administrator user is granted full
   access permissions when the operating system user name of the instance administrator user is
   "fsepuser". The following procedure applies when the user is logged in to the Windows server as
   "fsepuser":
   ```

   ```
   > takeown /f serverIdentifier.conf
   > icacls serverIdentifier.conf /reset
   > icacls serverIdentifier.conf /inheritance:r
   > icacls serverIdentifier.conf /grant fsepuser:F
   ```

7. Configure the Windows firewall.

   If the Windows firewall feature is to be enabled, you should enable the port number of Mirroring Controller that you specified in the network definition file in step 3. Refer to "E.2 Windows Firewall Settings" for details.

8. Register Mirroring Controller to the Windows service.

   Execute the mc_ctl command in the register mode.

   For the -P option of the mc_ctl command, specify the password of the operating system user who executes the command.

   Example)

```
> mc_ctl register -M D:\mcdir\inst1 -P ********
```

   Note

   When specifying the password in the -P option of the mc_ctl command, for security reasons, you should be careful not to allow other users to access it.

   Information

   You can use the mc_ctl command with the -S option to specify automatic start and stop of Mirroring Controller. Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for details.

   Using the service name specified in the mc_service_name parameter of *serverIdentifier*.conf in step 5, Mirroring Controller is registered to the Windows service.

   You can execute the sc qc command to check the registration status.

2.4.2 Creating, Setting, and Registering the Primary Server Instance

This section explains how to create, set, and register the primary server instance.

Note

- Mirroring Controller supports instances that are registered in the Windows service.

See

- Refer to "Client Authentication" in the PostgreSQL Documentation for information on the pg_hba.conf file.
- Refer to "A.1 Parameters Set on the Primary Server" for information on the postgresql.conf file.
- Refer to "mc_ctl" in Reference for information on the command.

Perform the following procedure:

1. Refer to "Setup" in the Installation and Setup Guide for Server, and then perform the Fujitsu Enterprise Postgres setup and create the Fujitsu Enterprise Postgres instance.

   - Use ASCII characters in the data storage destination directory.
   - When registering an instance to the Windows service, perform the settings required to enable Mirroring Controller to start and stop the instance. Execute the pg_ctl command with the following specified for the register mode:
     - For the service name of the -N option, specify the name set for the db_instance_service_name parameter in the server definition file
     - Specify "demand" for the -S option, so that the service does not start automatically on startup of the system

Note

- If degradation starts occurring due to an error during operations in database multiplexing mode, recovery is required for the standby server. There are some conditions to execute the pg_rewind command to recover the standby server. One of the conditions can be satisfied by enabling checksums when executing the initdb command. This is not mandatory. Refer to "4.1.1.1.3 Identify cause of error and perform recovery" for details.

- Do not configure the Windows service of a multiplexed instance to perform automatic start, as it is started by Mirroring Controller.

2. When using transparent data encryption, configure the encryption settings for the storage data.

   If you want to use the key management system as a keystore, set the connection information for the key management system and declare the master encryption key.

   Refer to "Protecting Storage Data Using Transparent Data Encryption" or "Using Transparent Data Encryption with Key Management Systems as Keystores" in the Operation Guide for details, and then configure the settings.

3. Add the following entry to the pg_hba.conf file to authenticate connections from the standby server.

   Copy the file to the standby server later.

```
# TYPE    DATABASE        USER        ADDRESS                 METHOD
  host    replication     fsep        standbyServerAddress    authenticationMethod
  host    replication     fsep        primaryServerAddress    authenticationMethod
```

   For the primary and standby server addresses, specify the IP address that will connect to the log transfer network.

   Additionally, all servers can be used as the primary server or the standby server, so add entries for the addresses of all servers that comprise the database multiplexing system.

Point

Setting an authentication method other than trust authentication

If the primary server becomes the standby server, to perform automatic authentication of connections to the primary server, create a password file (%APPDATA%\postgresql\pgpass.conf), and then specify a password for the replication database. Accordingly, the instance administrator operating system user and the user registered in the database will be the same, so you can verify that the connection was not made by an unspecified user. Additionally, the password that was set beforehand will be used in the authentication, so that the connection will be automatic.

Note

If trust authentication is set, all OS users who can log in to the primary server will be able to connect, and if one of these is a malicious user, then that user can corrupt the standby server data, or cause the job system to fail, by sending an erroneous transaction log. Therefore, decide on the authentication method according to the security requirements of the system using database multiplexing mode.

Refer to "Authentication Methods" in the PostgreSQL Documentation for details on the authentication methods that can be set.

4. Configure this setting to enable the instance administrator user of the primary server to connect as a database application.

   This setting enables the connection to the instance using the user name of the instance administrator user, so that Mirroring Controller can monitor instance errors. Configure this setting to enable the connection to the postgres database.

   - If password authentication is used

     In the db_instance_password parameter of the *serverIdentifier*.conf file, specify the password for the instance administrator user. This password is used to connect to the database instance. If a password is not specified in the db_instance_password parameter, the connection to the database instance from Mirroring Controller will fail, and it will not be possible to perform the process monitoring of the instance.

- If password authentication is not used

  There is no need to specify the password in the db_instance_password parameter.

  Even if the password for the instance administrator user is specified in the db_instance_password parameter, it will be ignored.

- If certificate authentication using SSL is used

  Specify connection parameters for SSL in the db_instance_ext_ pq_conninfo parameter and db_instance_ext_jdbc_conninfo parameter in the *serverIdentifier*.conf file. If the pasrameters are not specified, the connection to the database instance from Mirroring Controller will fail, and it will not be possible to perform the process monitoring of the instance. If certificate authentication using SSL is not performed, the parameters specification is not required.

An example of setting the authentication method is shown below.

```
# TYPE    DATABASE        USER        ADDRESS         METHOD
  host    postgres        fsep        127.0.0.1/32    authenticationMethod
```

Note

Mirroring Controller uses the PostgreSQL JDBC 4.2 driver to connect to the database instance. Therefore, for the authentication method, specify a method supported by the JDBC driver. If an authentication method not supported by the JDBC driver is specified, Mirroring Controller will fail to start. Refer to the PostgreSQL JDBC Driver Documentation for information on authentication methods supported by the JDBC driver.

5. To use database multiplexing mode, specify the parameters shown in the table below in the postgresql.conf file.

   The postgresql.conf file is copied when the standby server instance is created. Accordingly, set the required parameters in the standby server.

   To use database multiplexing mode, specify the parameters shown in the table below in the postgresql.conf file. After editing the postgresql.conf file, restart the instance.

Table 2.4 Parameters

Parameter	Content specified	Remarks
wal_level	replica or logical	Specify "logical" when logical decoding is also to be used.
max_wal_senders	2 or more	Specify "2" when building a Mirroring Controller cluster system. When additionally connecting asynchronous standby servers to the cluster system, add the number of simultaneous connections from these standby servers.
synchronous_standby_names	'*standbyServerName*'	Specify the name that will identify the standby server. Enclose the name in single quotation marks ('). Do not change this parameter while Mirroring Controller is running. Do not specify multiple names to this parameter as the Mirroring Controller can manage only one standby server.
hot_standby	on	Specify this to execute reference jobs on the standby server.
wal_keep_size	WAL save size (megabytes)	If a delay exceeding the value set in this parameter occurs, the WAL segment required later by the primary server may be deleted. Additionally, if you stop a standby server (for maintenance, for example), consider the stop time and set a value that will not cause the WAL segment to be deleted.

Parameter	Content specified	Remarks
		Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on estimating the WAL save size.
wal_log_hints	on	When using the pg_rewind command to recover a standby server, specify this parameter or enable checksums when executing the initdb command.
wal_sender_timeout	Timeout (milliseconds)	Specify the time period after which it is determined that an error has occurred in the transaction log transfer on the primary server. By aligning this value with the value for the database process heartbeat monitoring time, you can unify the time after which it is determined that an error has occurred.
archive_mode	on	Specify the archive log mode.
archive_command	'cmd /c ""installDir\\bin\\pgx_walcopy.cmd" "%p" "backupDataStorageDestinationDir\\archived_wal\\%f"""	Specify the command and storage destination to save the transaction log.
backup_destination	Backup data storage destination directory	Specify the name of directory where to store the backup data. Set the permissions so that only the instance administrator user can access the specified directory. Specify the same full path on all servers, so that the backup data of other servers can be used to perform recovery.
max_connections	Number of simultaneous client connections to the instance + superuser_reserved_connections	The value specified is also used to restrict the number of connections from client applications and the number of connections for the management of instances. Refer to "When an Instance was Created with the initdb Command" in the Installation and Setup Guide for Server, and "Connections and Authentication" in the PostgreSQL Documentation, for details.
superuser_reserved_connections	Add the number of simultaneous executions of mc_ctl status (*1) + 2	Specify the number of connections reserved for connections from database superusers. Add the number of connections from Mirroring Controller processes. Also reflect the added value in the max_connections parameter.
wal_receiver_timeout	Timeout (milliseconds)	Specify the time period after which it is determined that an error has occurred when the transaction log was received on the standby server. By aligning this value with the value for the heartbeat monitoring time of the database process, you can unify the time after which it is determined that an error has occurred.
restart_after_crash	off	If "on" is specified, or the default value is used for this parameter, behavior equivalent to restarting Fujitsu Enterprise Postgres, including crash recovery, will be performed when some server processes end abnormally. However, when database multiplexing monitoring is used, a failover will occur after an error is detected when some server processes end abnormally, and the restart of those

Parameter	Content specified	Remarks
		server processes is forcibly stopped. Specify "off" to prevent behavior such as this from occurring for no apparent reason.
synchronous_commit	on or remote_apply	Specify up to what position WAL send is to be performed before transaction commit processing returns a normal termination response to a client. The recommended value is on or remote_apply to prevent data loss caused by operating system or server down immediately after a switch or switch.
recovery_target_timeline	latest	Specify "latest" so that the new standby server (original primary server) will follow the new primary server when a switch occurs. This parameter is required when the original primary server is incorporated as a new standby server after the primary server is switched.

*1: Number of simultaneous executions of the mc_ctl command in the status mode.

2.4.3 Starting Mirroring Controller on the Primary Server

This section explains how to start Mirroring Controller on the primary server.

When the arbitration server is used for automatic degradation, start the Mirroring Controller arbitration process on the arbitration server in advance.

1. Start the Mirroring Controller process.

   Enabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode.

   ```
   Example)
   ```

   ```
   > mc_ctl start -M D:\mcdir\inst1
   ```

   Disabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

   ```
   Example)
   ```

   ```
   > mc_ctl start -M D:\mcdir\inst1 -F
   ```

Note

- When the arbitration server is used for automatic degradation, the database server must connect to the arbitration server, and as a result, Mirroring Controller startup may take longer than when the arbitration server is not used.
- If the parameter for heartbeat monitoring of operating systems or servers set by the abitration server is greater than parameter for heartbeat monitoring of operating systems and servers of the Mirroring Controller, the Mirroring Controller may fail to start. In this case, check the contents of the message notification and review the parameters for heartbeat monitoring of operating systems or servers for the arbitration server or Mirroring Controller.
- If the heartbeat_error_action parameter in *serverIdentifier*.conf is set to "message", even if automatic switch/disconnection is enabled and Mirroring Controller is started, only message output is performed when a heartbeat abnormality is detected during heartbeat monitoring of operating systems and servers - switch/disconnection is not performed.
- Mirroring Controller startup usually fails if the standby server is mistakenly started as the primary server or if the old primary server is not recovered after the switch and is then mistakenly started as the primary server. However, if the admin network is

disconnected, then startup does not fail, and both servers may become primary servers. Therefore ensure that the admin network is connected before starting Mirroring Controller.

Point

- The mc_ctl command fails if the Mirroring Controller arbitration process has not been started on the arbitration server when the arbitration server is used for automatic degradation. However, if the Mirroring Controller arbitration process cannot be started in advance, it can be started by specifying the --async-connect-arbiter option in the mc_ctl command.
- After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

2. Obtain the backup.

   Use the pgx_dmpall command to collect the backup.

2.5 Setting Up the Standby Server

This section explains how to set up the standby server.

2.5.1 Setting Up Database Multiplexing Mode on the Standby Server

This section explains how to set up database multiplexing mode on the standby server.

In database multiplexing, the files that are required for operations are managed in the Mirroring Controller management directory.

There is one Mirroring Controller management directory for each instance.

Note

- Do not place the Mirroring Controller management directory in a directory managed by Fujitsu Enterprise Postgres, otherwise it may be deleted by mistake with the directories managed by Fujitsu Enterprise Postgres, and an old version of files may be restored.
- When creating a standby server for a large database, stop job system operations, specify a large value for the wal_keep_size parameter, or use replication slots.
  This is because WALs generated after the standby server is built using the pg_basebackup command, but before it is started, need to be retained. However, the number of WAL segments that can be retained is constrained by the wal_keep_size parameter.
  Additionally, setting the wal_keep_size parameter requires consideration regarding stabilization of the database multiplexing mode (refer to "2.11.1 Tuning to Stabilize the Database Multiplexing Mode" for details).

See

- Refer to "Preparing Directories for Resource Deployment" in the Installation and Setup Guide for Server for details on the directories that are managed by Fujitsu Enterprise Postgres.
- Refer to "pg_basebackup" in "Reference" in the PostgreSQL Documentation for information on the pg_basebackup command.
- Refer to "mc_ctl" in Reference for information on the command.
- Refer to "Appendix A Parameters" for details on each parameter to be edited for the setup.
- Refer to "Replication Slots" in the PostgreSQL Documentation for information on replication slots.

Perform the following procedure:

1. Log in to the standby server.

2. Create the Mirroring Controller management directory that will store the files required by database multiplexing.

   Use ASCII characters in the Mirroring Controller management directory.

   Additionally, grant "Write" permission to the instance administrator user for the Mirroring Controller management directory.

3. Copy, and then deploy, the network.conf file of the primary server.

   Copy the network.conf file that was defined in the primary server setup, and deploy it to the Mirroring Controller management directory of the standby server.

   Register the port number of the standby server that was specified in the network.conf file in the services file, because there are programs, such as WebAdmin, that search an available port number using the services file.

   Register any name as the service name.

4. Change the access permissions for the network.conf file.

   Set read and write permissions for the instance administrator user only. If users other than the instance administrator user are granted access permissions, the mc_ctl command will not work. Accordingly, users other than the instance administrator user are prevented from operating Mirroring Controller.

   ```
   Example)
   The following is an execution example, in which the instance administrator user is granted full
   access permissions as the owner when the operating system user name of the instance administrator
   user is "fsepuser". The following procedure applies when the user is logged in to the Windows
   server as "fsepuser".
   ```

   ```
   > takeown /f network.conf
   > icacls network.conf /reset
   > icacls network.conf /inheritance:r
   > icacls network.conf /grant fsepuser:F
   ```

5. Copy, and then deploy, the *serverIdentifier*.conf file of the primary server.

   Copy the *serverIdentifier*.conf file that was defined in the primary server setup, and deploy it to the Mirroring Controller management directory of the standby server.

   Note

   If the primary server and standby server are to be built within the same server, change the following parameters in the *serverIdentifier*.conf file, ensuring that the names are not duplicated with those on the primary server:

   - db_instance_service_name (registered service name of the Fujitsu Enterprise Postgres instance)
   - mc_service_name (registered service name of Mirroring Controller)
   - event_source (event source name)

6. Change the access permissions for the *serverIdentifier*.conf file.

   Set read and write permissions for the instance administrator user only. If users other than the instance administrator user are granted access permissions, the mc_ctl command will not work.

   ```
   Example)
   The following is an execution example, in which the instance administrator user is granted full
   access permissions as the owner when the operating system user name of the instance administrator
   user is "fsepuser". The following procedure applies when the user is logged in to the Windows
   server as "fsepuser":
   ```

   ```
   > takeown /f serverIdentifier.conf
   > icacls serverIdentifier.conf /reset
   > icacls serverIdentifier.conf /inheritance:r
   > icacls serverIdentifier.conf /grant fsepuser:F
   ```

7. Configure the Windows firewall.

   If the Windows firewall feature is to be enabled, you should enable the port number of Mirroring Controller that you specified in the network definition file in step 3. Refer to "E.2 Windows Firewall Settings" for details.

8. Register Mirroring Controller to the Windows service.

   Execute the mc_ctl command in the register mode.

   For the -P option of the mc_ctl command, specify the password of the operating system user who executes the command.

   Example)

```
> mc_ctl register -M D:\mcdir\inst1 -P ********
```

Note

When specifying the password in the -P option of the mc_ctl command, for security reasons, you should be careful not to allow other users to access it.

Information

You can use the mc_ctl command with the -S option to specify automatic start and stop of Mirroring Controller. Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for details.

Using the service name specified in the mc_service_name parameter of *serverIdentifier*.conf in step 5, Mirroring Controller is registered to the Windows service.

You can execute the sc qc command to check the registration status.

2.5.2 Creating, Setting, and Registering the Standby Server Instance

This section explains how to create, set, and register the standby server instance.

Note

Mirroring Controller supports instances that are registered in the Windows service.

See

- Refer to "Appendix A Parameters" for details on each parameter.
- Refer to "mc_ctl" in Reference for information on the command.

Perform the following procedure:

1. Prepare for setup.

   Refer to "Preparations for Setup" in the Installation and Setup Guide for Server for information on the preparatory tasks to be performed before creating an instance on the standby server.

   Note

   If the primary server and standby server are to be built within the same server, perform preparation to ensure that the event source names of Fujitsu Enterprise Postgres are not duplicated with that of the primary server.

2. When using transparent data encryption, configure the encryption settings for the storage data.

   Refer to "Protecting Storage Data Using Transparent Data Encryption" or "Using Transparent Data Encryption with Key Management Systems as Keystores" in the Operation Guide for details, and then configure the settings.

3. Execute the pg_basebackup command to create a copy of the primary server instance on the standby server.

   Example)

```
> pg_basebackup -D D:\database\inst1 -X fetch --waldir=E:\transaction\inst1 --progress --verbose
-R --dbname="application_name=standbyServerName" -h primaryServerIpAddress -p
primaryServerPortNumber
```

Note

- Use the pg_basebackup command with the -R option to create a standby.signal file. If you do not create the standby.signal file, the Mirroring Controller cannot be started as a standby server.
- If using a method that requires password authentication for connections to the primary server, you will need to ensure that authentication is performed automatically. If the -R option is specified for the pg_basebackup command and the password parameter is specified for the --dbname option, the pg_basebackup command will set the password in the primary_conninfo parameter in postgresql.auto.conf file, enabling connections to be performed automatically.

  If a password is not set in the primary_conninfo parameter in postgresql.auto.conf file, it will be necessary to create a password file (%APPDATA%postgresql\pgpass.conf), and then specify a password for the replication database.
- The primary_conninfo parameter should not be set in the postgresql.conf file, but only in the postgresql.auto.conf file using the pg_basebackup command.
- When executing the pg_basebackup command, consider the following for collection of transaction logs.
  - When "fetch" is specified for the -X option of the command

    Transaction logs are collected at the end of the backup, so it is necessary to ensure that transaction logs that occur during backup are not deleted from the primary server. Therefore, allow for a sufficient value for the wal_keep_size parameter in postgresql.conf.
  - When the -X option is omitted or "stream" is specified for the -X option of the command

    Transaction logs are streamed, so when Mirroring Controller is running on the primary server, the connection is changed to a synchronous standby server on detection of a streaming replication connection using this command. Therefore, if a job has started on the primary server, the primary server will be impacted, therefore execute this command after stopping only the Mirroring Controller process on the primary server.

See

Refer to "Hot Standby" in the PostgreSQL Documentation for information on the standby.signal file.

4. Set the parameters shown in the table below in the postgresql.conf file.

Table 2.5 Parameters

Parameter	Content specified	Remarks
synchronous_standby_names	'*primaryServerName*'	Required after switching the primary server and then changing the original primary server to the new standby server. Enclose the name in single quotation marks ('). Do not change this parameter while Mirroring Controller is running. Do not specify multiple names to this parameter as the Mirroring Controller can manage only one standby server.

5. Register an instance to the Windows service.

   Refer to "Creating an Instance" in the Installation and Setup Guide for Server for information on how to register an instance to the Windows service. Note that you should execute the pg_ctl command with the following specified for the register mode to enable Mirroring Controller to start and stop an instance:

   - For the service name of the -N option, specify the name set for the db_instance_service_name parameter in the server definition file
   - Specify "demand" for the -S option, so that the service does not start automatically on startup of the system

   Note

   - Do not configure the Windows service of a multiplexed instance to perform automatic start, as it is started by Mirroring Controller.
   - If the primary server and standby server are to be built within the same server, ensure that the registered service name of the Fujitsu Enterprise Postgres instance is not duplicated with that of the primary server.

2.5.3 Starting Mirroring Controller on the Standby Server

This section explains how to start Mirroring Controller on the standby server.

When the arbitration server is used for automatic degradation, start the Mirroring Controller arbitration process on the arbitration server in advance.

1. After ensuring that the Mirroring Controller process of the primary server has started, start Mirroring Controller on the standby server.

   Enabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode with the -f option specified. This action enables automatic switch/disconnection.

   If you start Mirroring Controller and the instance without specifying the -f option, automatic switch/disconnection will not be enabled. To enable both, start Mirroring Controller and then execute the mc_ctl command in enable-failover mode or restart Mirroring Controller with the -f option specified.

   Example)

   ```
   > mc_ctl start -M D:\mcdir\inst1
   ```

   Disabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

   Example)

   ```
   > mc_ctl start -M D:\mcdir\inst1 -F
   ```

2. Check the status of the Mirroring Controller process.

   As the instance administrator user, execute the mc_ctl command in status mode. Ensure that "mirroring status" is switchable.

   Example)

   ```
   > mc_ctl status -M D:\mcdir\inst1
   ```

Note

- When the arbitration server is used for automatic degradation, the time required for the database server to connect to the arbitration server is added on. Therefore, Mirroring Controller startup may take longer than when the arbitration server is not used.
- If the parameter for heartbeat monitoring of operating systems or servers set by the abitration server is greater than parameter for heartbeat monitoring of operating systems and servers of the Mirroring Controller, the Mirroring Controller may fail to start. In this

case, check the contents of the message notification and review the parameters for heartbeat monitoring of operating systems or servers for the arbitration server or Mirroring Controller.

- If the heartbeat_error_action parameter in *serverIdentifier*.conf is set to "message", even if automatic switch/disconnection is enabled and Mirroring Controller is started, only message output is performed when a heartbeat abnormality is detected during heartbeat monitoring of operating systems and servers - switch/disconnection is not performed.
- Mirroring Controller startup usually fails if the standby server is mistakenly started as the primary server or if the old primary server is not recovered after the switch and is then mistakenly started as the primary server. However, if the admin network is disconnected, then startup does not fail, and both servers may become primary servers. Therefore, ensure that the admin network is connected before starting Mirroring Controller.

Point

- The mc_ctl command fails if the Mirroring Controller arbitration process has not been started on the arbitration server when the arbitration server is used for automatic degradation. However, if the Mirroring Controller arbitration process cannot be started in advance, it can be started by specifying the --async-connect-arbiter option in the mc_ctl command.
- After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

2.6 Creating a User Exit for a Database Server

This section explains how to create a user exit for a database server.

The user command types explained below can be used as user exits. These commands are called by Mirroring Controller management processes.

The user can create user exits as required.

Specify the user commands that were created for the parameters in the server configuration file of the database server. Refer to "A.4.1 Server Configuration File for the Database Servers" for information on these parameters.

User command types

- Fencing command

  This user command performs fencing if Mirroring Controller performs arbitration processing and determines that a database server is unstable.

- Arbitration command

  This user command performs arbitration processing in lieu of the arbitration server when there is no arbitration server.

- State transition commands

  These user commands are called when Mirroring Controller performs state transition of a database server.
  It includes the following types:

  - Post-switch command

    This user command is called after a promotion from standby server to primary server.

  - Pre-detach command

    This user command is called before the standby server is disconnected from a cluster system.
    If the pre-detach command is specified on both the primary server and standby server, it is called first on the standby server and then on the primary server.

    If the settings are configured to forcibly stop the instance on the standby server when the standby server is disconnected, the pre-detach command is called on the standby server and then the instance on the standby server is stopped.

- Post-attach command

  This user command is called after the standby server has been attached to a cluster system.
  If the post-attach command is specified on both the primary server and standby server, it is called first on the primary server and then on the standby server.

Point

When the arbitration server is used for automatic degradation and the requirements can be satisfied using the fencing command on the arbitration server only, the fencing command on the database server is not required. In addition, if the requirements can be satisfied using the fencing command on the database server only, create a fencing command on the arbitration server for termination processing only (without implementation).

Table 2.6 Availability of user commands, and database server calling the command

User command	Operation when a heartbeat abnormality is detected using operating system or server heartbeat monitoring				Database server calling the command	
	Message output	Automatic degradation by arbitration server	Automatic degradation by arbitration command	Unconditional automatic degradation	Primary server	Standby server
Fencing command	Y (*1)	Y (*2)	R	N	Y	Y
Arbitration command	N	N	R	N	Y	Y
Post-switch command	Y	Y	Y	Y	Y	N
Pre-detach command	Y	Y	Y	Y	Y	Y (*3)
Post-attach command	Y	Y	Y	Y	Y	Y (*3)

R: Required

Y: Can be used

N: Cannot be used

*1: Called only when the mc_ctl command is used to execute forced switching or forced disconnection.

*2: Creation of a fencing command on a database server is optional, but it must be created on the arbitration server.

*3: If message output or unconditional automatic degradation is selected, this command is called only from the primary server.

See

Refer to "Appendix C User Commands" for information on the interface for each user command.

2.7 Confirming the Streaming Replication Status

Before performing the setup of the database multiplexing mode, ensure that the prerequisite streaming replication feature has been set up correctly.

Perform the following procedure:

1. On the primary server, ensure that single-row searches can be performed using the pg_stat_replication statistics view.

   An example output of the psql command is shown below.

   ```
   Example)
   ```

```
postgres=# select * from pg_stat_replication;
-[ RECORD 1 ]----+------------------------------
pid              | 10651
usesysid         | 10
usename          | fsep
application_name | standby
client_addr      | 192.0.2.210
client_hostname  |
client_port      | 55098
backend_start    | 2022-03-23 11:17:49.628793+09
backend_xmin     |
state            | streaming
sent_lsn         | 0/3000060
write_lsn        | 0/3000060
flush_lsn        | 0/3000060
replay_lsn       | 0/3000060
write_lag        |
flush_lag        |
replay_lag       |
sync_priority    | 1
sync_state       | sync
reply_time       | 2022-03-23 11:23:27.703366+09
```

2. Confirm the search results of step 1.

   Ensure that the connection established with the intended standby server is in synchronous mode.

   Table 2.7 Items to be checked

Item	Required value
application_name	Value specified for synchronous_standby_names parameter in the postgresql.conf file of the primary server.
client_addr	IP address of the standby server.
state	"streaming".
sync_state	"sync".

See

- Refer to "The Statistics Collector" in "Server Administration" in the PostgreSQL Documentation for information on the pg_stat_replication statistics view.
- Note that the pg_stat_replication statistics view may change in the future.

2.8 Checking the Connection Status

This section explains how to check the connection status from a database server or the arbitration server.

2.8.1 Checking the Connection Status on a Database Server

This section explains how to use a database server to check the connection status of the Mirroring Controller arbitration process and the Mirroring Controller process on the primary server and standby server.

Perform the following procedure:

1. On the primary server and standby server, execute the mc_ctl command in status mode with the --arbiter option specified.

   ```
   Example)
   The mc_ctl command is executed with the --arbiter option specified, and the status is output.
   ```

```
> mc_ctl status --arbiter -M D:\mcdir\inst1

arbiter_id   host            status
-------------------------------------
arbiter      192.0.3.120     online
```

2. On the primary server and standby server, check the result displayed by executing the mc_ctl command in status mode in step 1.

   Items to be checked

   Check that the output status" is "online".

See

Refer to the Reference for information on the mc_ctl command.

2.8.2 Checking the Connection Status on the Arbitration Server

This section explains how to use the arbitration server to check the connection status of the Mirroring Controller arbitration process and the Mirroring Controller process on the primary server and standby server.

Perform the following procedure:

1. Execute the mc_arb command in status mode on the arbitration server.

   The example below executes the mc_arb command, and shows the status.

   Linux

   Example)

```
$ mc_arb status -M /mcarb_dir/arbiter1

server_id    host            status
-------------------------------------
server1      192.0.3.100     online
server2      192.0.3.110     online
```

   Windows

   Example)

```
> mc_arb status -M D:\mcarb_dir\arbiter1

server_id    host            status
-------------------------------------
server1      192.0.3.100     online
server2      192.0.3.110     online
```

2. Check the result displayed by executing the mc_arb command in step 1.

   Items to be checked

   Check that the output status is "online" on both lines.

See

Refer to the Reference for information on the mc_arb command.

2.9 Creating Applications

This section explains how to create applications using database multiplexing, and points that should be noted when you create the applications.

2.9.1 Application Connection Server Settings

If database multiplexing is used and a failover occurs, it will be necessary to switch the application connection server. Accordingly, use the application connection switch feature to create applications.

Refer to "Application Connection Switch Feature" in the Application Development Guide for details.

2.10 Checking the Behavior

To check if the environment setup was performed correctly, start the application and then check the behavior of the switch and rebuild.

2.11 Tuning

This section explains how to tune database multiplexing mode.

2.11.1 Tuning to Stabilize the Database Multiplexing Mode

When large amounts of data are updated, the write-to load for the database will become great, and the multiplexing state may become unstable.

Accordingly, by editing the parameters below in the postgresql.conf file, a stable multiplexing state can be maintained. Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on transaction log space requirements.

Table 2.8 Parameters

Parameter	Content
wal_keep_size	Refer to "2.4.2 Creating, Setting, and Registering the Primary Server Instance" for details.
max_wal_size	The transaction log is written out according to the checkpoint trigger. If a transaction log with the capacity of the value specified in this parameter is generated, the checkpoint will be executed. If a large value is specified in this parameter, the time required for crash recovery will increase. If a small value is specified in this parameter, many checkpoints will be generated, which will affect the performance of the applications that connect to the primary server.

2.11.2 Tuning to Stabilize Queries on the Standby Server

Queries made using reference jobs on the standby server may be canceled by jobs executed on the primary server.

To reduce the possibility of a job being canceled, specify as large a value as possible for the max_standby_archive_delay parameter in the postgresql.conf file.

- Refer to "Handling Query Conflicts" in the PostgreSQL Documentation for details.
- Refer to "Standby Servers" in the PostgreSQL Documentation for details on the max_standby_archive_delay parameter.

2.11.3 Tuning to Stabilize Queries on the Standby Server (when Performing Frequent Updates on the Primary Server)

If jobs are updated on the primary server regularly and frequently, it will be easy for the query made by the reference job on the standby server to be canceled. In this case, edit one of the postgresql.conf file parameters shown in the table below.

Table 2.9 Parameters

Parameter	Description
hot_standby_feedback	When "on" is set, the deletion (vacuum) of the data area that was deleted or updated on the primary server is suppressed. Accordingly, the query on the standby server will not be canceled. (*1)
vacuum_defer_cleanup_age	The deletion (vacuum) of the data area that was deleted or updated on the primary server is delayed until the specified number of transactions is processed. Accordingly, the probability that the query on the standby server will be canceled decreases.

*1: Because the vacuum is delayed, the data storage destination disk space of the primary server comes under pressure.
Additionally, if there is conflict between accesses and queries executed on the standby server, transaction logs indicating this conflict will be transferred.
Accordingly, specify as large a value as possible for the max_standby_archive_delay parameter so that access conflicts do not occur.

See

- Refer to "Standby Servers" in the PostgreSQL Documentation for details on the hot_standby_feedback parameter.
- Refer to "Primary Server" in the PostgreSQL Documentation for details on the vacuum_defer_cleanup_age parameter.

2.11.4 Tuning for Optimization of Degradation Using Abnormality Monitoring

Mirroring Controller uses a monitoring method that outputs an error if the timeout or number of retries is exceeded when accessing resources targeted for monitoring. Setting inappropriate values in these settings may lead to misdetection or a delay in automatic degradation, so you must design these values appropriately.

For example, the following type of issue occurs if the tuning related to abnormality monitoring is not performed appropriately.

- If the timeout is too short

  Results in redundant degradation and availability falls.

- If the timeout is too long

  It takes longer for automatic degradation to be performed even when an error affecting operational continuity occurs, potentially causing downtime.

You can optimize degrading operation by editing the values for the parameters in the server configuration file described below in accordance with the system. Refer to "A.4 Server Configuration File" for information on how to edit these parameters.

2.11.4.1 Tuning for Abnormality Monitoring of the Operating System or Server

Tuning for abnormal monitoring of the operating system or server depends on the operation when heartbeat abnormality is detected by the heartbeat monitoring of operating systems or servers.

See

Refer to "1.1.1 Monitoring Using Database Multiplexing Mode" for the operation when heartbeat abnormality is detected in the the heartbeat monitoring of operating systems or servers.

2.11.4.1.1 Tuning Abnormality Monitoring for Operations that Use an Arbitration Server for Automatic Degeneration

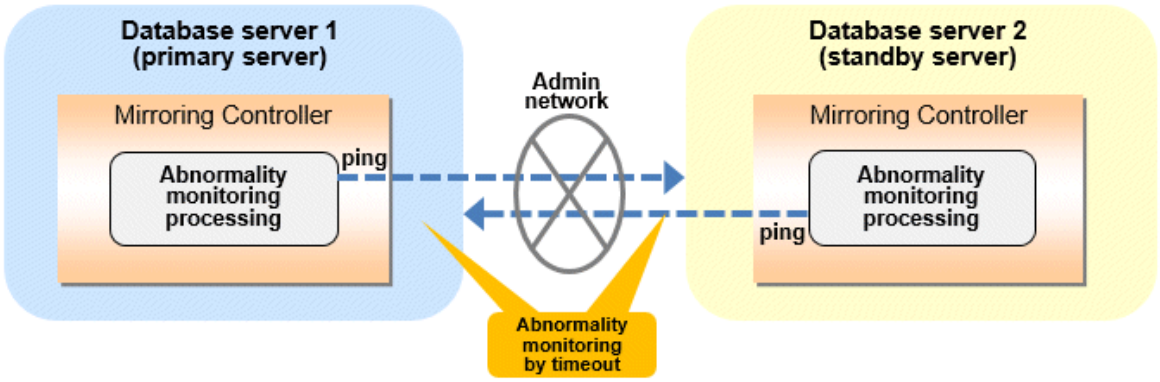
In an operation that use an arbitration server for automatic degeneration, the database server is periodically monitored for abnormalities so that the Mirroring Controller arbitration process can immediately respond to an arbitration request from the Mirroring Controller.The automatic degradation using the arbitration server can optimize the time from error detection to automatic degradation of the operating systems or servers by editing the following parameters.

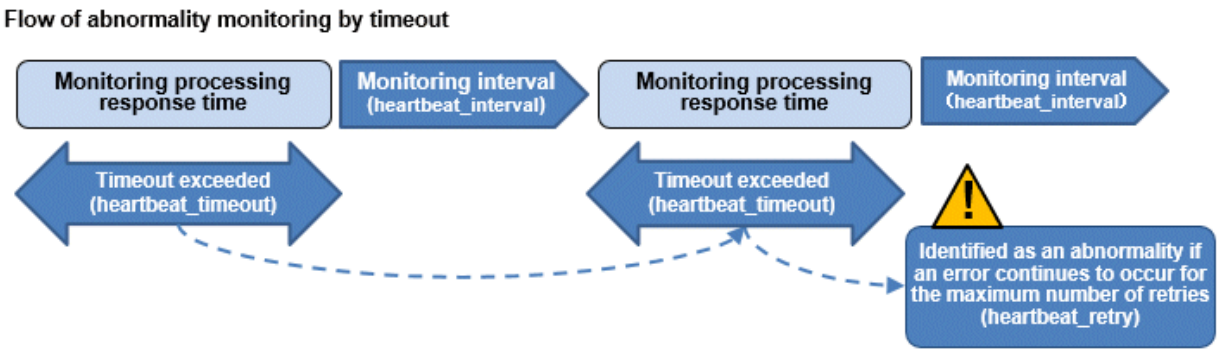
- Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server
- Parameters for the abnormality monitoring of the operating system or server in the arbitration configuration file
- Parameters for the arbitration processing and fencing

Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server

Table 2.10 Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server

Parameter	Description
Abnormality monitoring interval (heartbeat_interval)	Mirroring Controller is configured so that abnormality monitoring does not place a load on the system. This parameter does not normally need to be set. (The default is 800 milliseconds.)
Abnormality monitoring timeout (heartbeat_timeout)	Take into account the time during which a load is placed continuously on the server or admin network performance. For example, it is envisaged that this parameter will be used in situations such as when performing high-load batch jobs or when a large number of online jobs occur continuously and concurrently. (The default is 1 second.)
Abnormality monitoring retries (heartbeat_retry)	This parameter can be set when needing a safety value for situations in which the value specified for heartbeat_timeout is exceeded, for example, when using systems with fluctuating loads, however, this parameter does not normally need to be set. (The default is 2 times.)





The expression for calculating the time required to detect an abnormality by Mirroring Controller is shown below.

```
Abnormality detection time of Mirroring Controller = ( heartbeat_timeout(seconds) +
heartbeat_interval(milliseconds) / 1000 ) x ( heartbeat_retry(number of times) + 1)
```

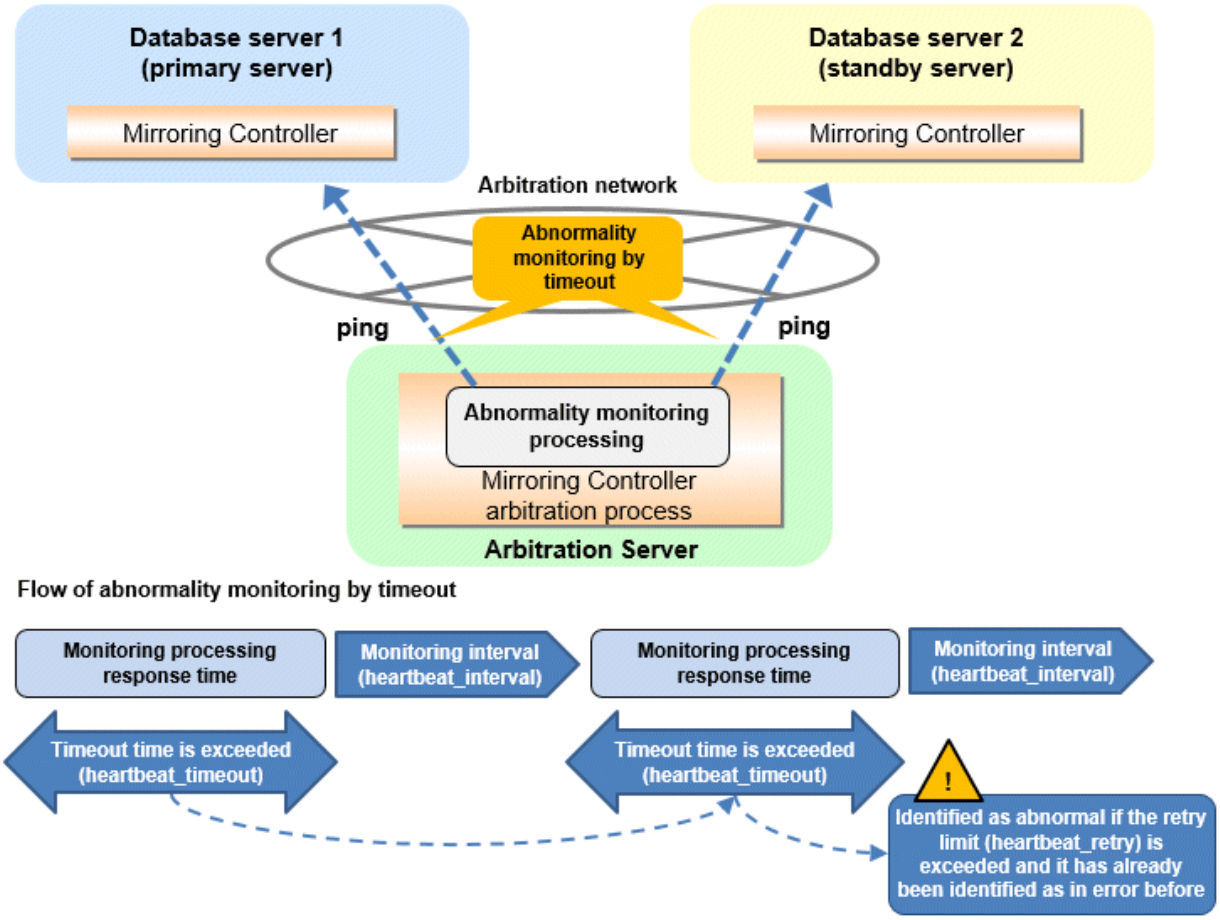
The abnormality detection time when the default value is used is shown below.

```
Abnormality detection time of Mirroring Controller = ( 1 + 800 / 1000 ) x ( 2 + 1 )
            = 5.4(seconds)
```

Parameters for the abnormality monitoring of the operating system or server in the arbitration configuration file

Table 2.11 Parameters for the abnormality monitoring of the operating system or server in the arbitration configuration file

Parameter	Description
Abnormality monitoring interval (heartbeat_interval)	Mirroring Controller arbitration process is configured so that abnormality monitoring does not place a load on the system. This parameter does not normally need to be set. (The default is the value set in heartbeat_interval in the server configuration file of the database server.) (milliseconds).
Abnormality monitoring timeout (heartbeat_timeout)	Take into account the time during which a load is placed continuously on the server and arbitration network capabilities. (The default is the value set in heartbeat_timeout in the server configuration file of the database server.) (seconds).
Abnormality monitoring retries (heartbeat_retry)	This parameter can be set when needing a safety value for situations in which the value specified for heartbeat_timeout is exceeded, for example, when using systems with fluctuating loads, however, this parameter does not normally need to be set. (The default is the value set in heartbeat_retry in the server configuration file of the database server.) (number of times)



The expression for calculating the time required to detect an abnormality by Mirroring Controller arbitration process is shown below.

```
Abnormality detection time of Mirroring Controller arbitration process = ( heartbeat_timeout(seconds)
+ heartbeat_interval(milliseconds) / 1000 ) x ( heartbeat_retry(number of times) + 1)
```

The abnormality detection time when the default value is used is shown below.

```
Abnormality detection time of Mirroring Controller arbitration process = ( 1 + 800 / 1000 ) x ( 2 + 1 )
            = 5.4(seconds)
```

Point

The abnormality detection time of the operation for automatic degradation using the arbitration server can be calculated as follows.

```
Abnormality detection time = Max(Abnormality detection time by Mirroring Controller, Abnormality
detection time by Mirroring Controller arbitration process)
```

Note

If the heartbeat_interval is set in the arbitration configuration file, the relationship between the parameter for operating system or server abnormality monitoring specified in the server configuration file of the database server file and the heartbeat_interval of the arbitration configuration file must satisfy the following relational expression.

```
 Heartbeat_interval in the arbitration configuration file (milliseconds) / 1000 ) <
( heartbeat_timeout(seconds) + heartbeat_interval(milliseconds) / 1000 ) * heartbeat_retry(number of
times) + heartbeat_timeout(seconds)
```

Parameters for the arbitration processing and fencing

Table 2.12 Parameters for the arbitration processing and fencing

Parameter	Description
Arbitration processing timeout (arbitration_timeout in the server configuration file of the database server)	Take into account the time to perform arbitration processing on the Mirroring Controller arbitration process. The value must be greater than or equal to abnormality detection time of Mirroring Controller arbitration process + fencing_command_timeout in the arbitration configuration file (seconds).
Fencing timeout (fencing_command_timeout in the arbitration configuration file)	Take into account the time to execute the fencing command (seconds).

Flow from the abnormality detection to the automatic degeneracy

When performing automatic degradation using the arbitration server, the flow from the abnormality detection in the operating system or server to the occurrence of automatic degeneracy and the parameters is shown below.

<table>
<tr><th>Flow from the abnormality detection to the automatic degeneracy</th><th>Description</th><th colspan="2">Parameter</th></tr>
<tr><td>(1) Abnormality detection</td><td>Mirroring Controller detect the database server operating system or server errors.</td><td colspan="2">Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server</td></tr>
<tr><td>(2) Arbitration request</td><td>Mirroring Controller that detect the operating system or server error asks the Arbitration Server to check the status of the other server's operating system or server.</td><td>-</td><td rowspan="3">arbitration_timeout in the server configuration file of the database server</td></tr>
<tr><td>(3) Arbitration processing</td><td>The Mirroring Controller arbitration process checks the status of the other server's operating system or server. However, if the result of the operating system or server abnormality monitoring by the arbitration server has been determined before the arbitration request from the Mirroring Controller of the database server, this process is not performed.</td><td>Parameters for the abnormality monitoring of the operating system or server in the arbitration configuration file</td></tr>
<tr><td>(4) Fencing</td><td>If the Mirroring Controller arbitration process determines that the other server is an abnomaly of the operating system or server, it fences the other server and isolates it from the cluster system. If the Mirroring Controller arbitration process determines that the operating system or server status is normal, this process and the (6) are not performed.</td><td>fencing_command_timeout in the arbitration configuration file</td></tr>
<tr><td>(5) Return of the arbitration results</td><td>Returns the results of the arbitration to the Mirroring Controller of the database server that requested the arbitration.</td><td colspan="2">-</td></tr>
</table>

Flow from the abnormality detection to the automatic degeneracy	Description	Parameter
(6) Automatic degradation	The automatic degradation is performed. If fencing fails in (4), this procedure is not performed.	-

-: No associated parameters

Note

If the fencing_command parameter is specified in the server configuration file of the database server, the fencing command is invoked on the database server if fencing is successful on the arbitration server. In that case, add the value of the fencing_command_timeout parameter in the server configuration file of the database server to the estimate.

Figure 2.1 When the Mirroring Controller on the primary server detects an operating system or server error

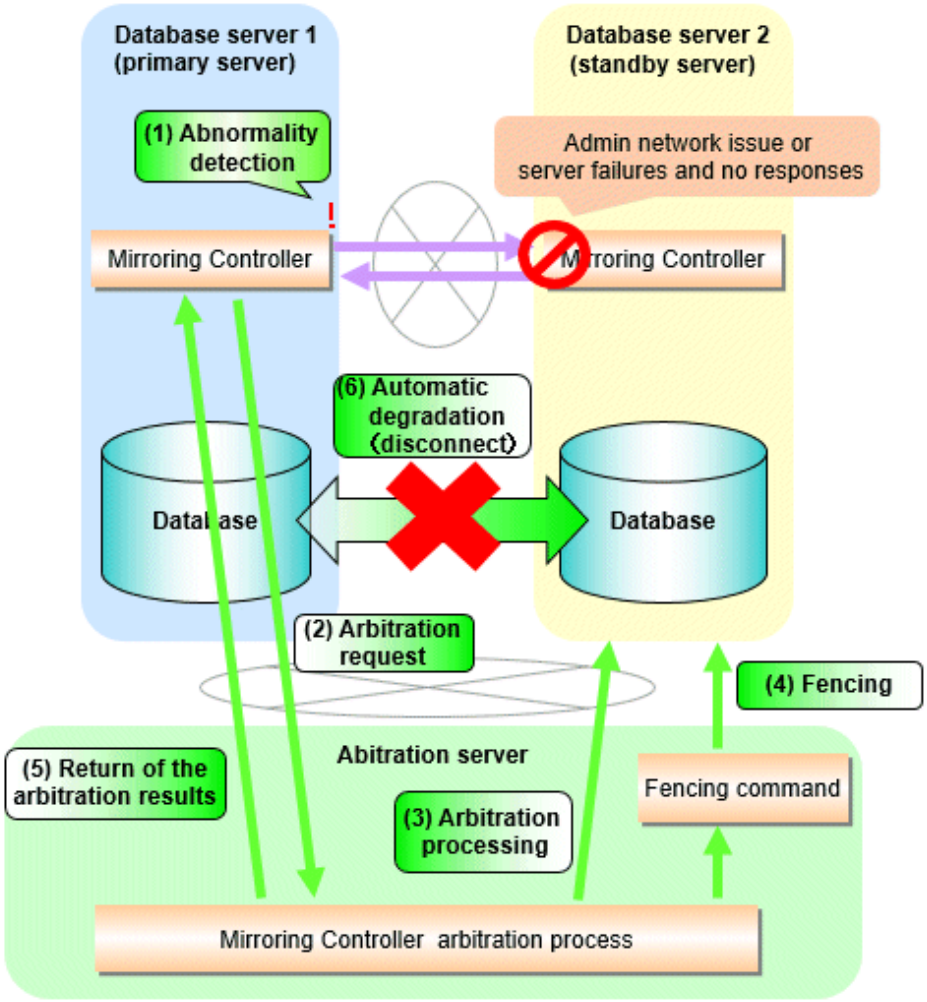
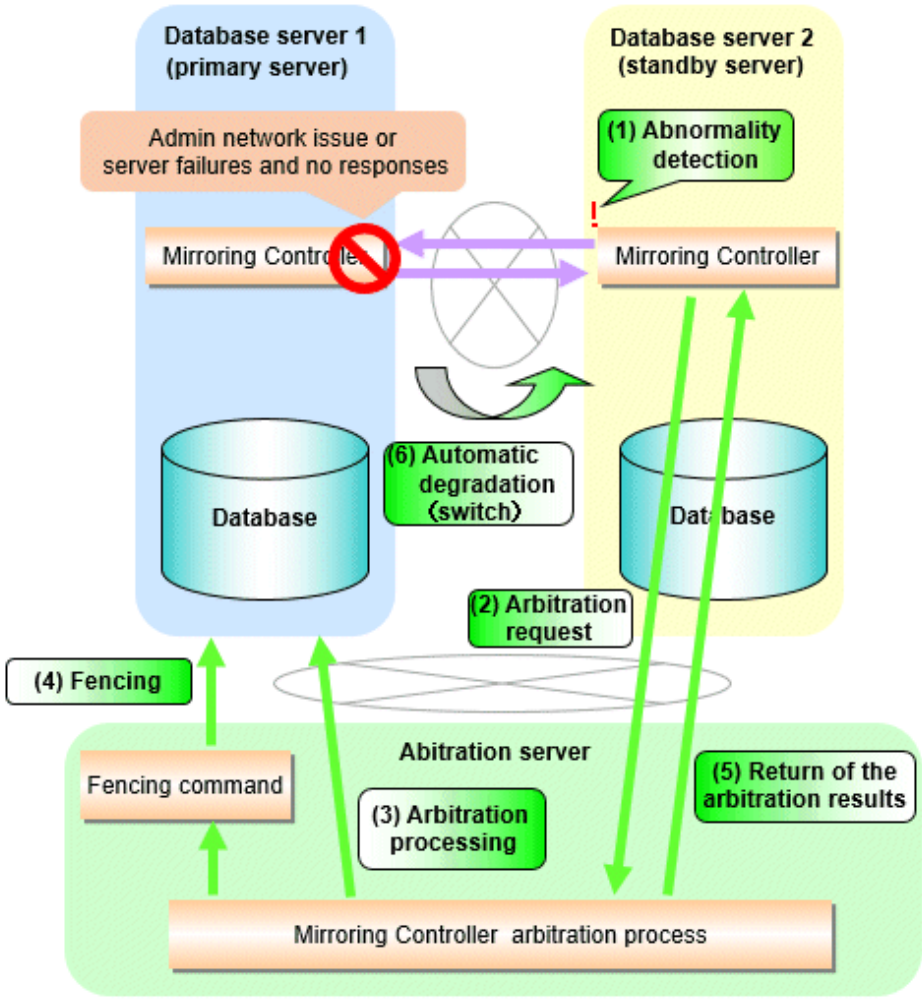


Figure 2.2 When the Mirroring Controller on the standby server detects an operating system or server error



2.11.4.1.2 Tuning Abnormality Monitoring for Operations that Perform Automatic Degeneration by Calling a User Exit that Determines Degeneration

In an operation that perform automatic degeneration by calling a user exit that determines degeneration, you can optimize the time from operating system or server abnormality detection to automatic degradation by editing the operating system or server abnormality monitoring parameters and parameters related to arbitration processing and fencing in the server configuration file of the database server. Refer to "Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server" for information on the operating system or server abnormality monitoring parameters in the server configuration file of the database server.

Table 2.13 Parameters for the arbitration processing and fencing

Parameter	Description
Arbitration processing timeout (arbitration_command_timeout)	Take into account the time to execute the arbitration command(seconds).
Fencing timeout (fencing_command_timeout)	Take into account the time to execute the fencing command (seconds).

Flow from the abnormality detection to the automatic degeneracy

When performing automatic degradation by calling a user exit that determines degeneration, the flow from the abnormality detection in the operating system or server to the occurrence of automatic degeneracy and the parameters is shown below.

Flow from the abnormality detection to the automatic degeneracy	Description	Parameter
(1) Abnormality detection	Mirroring Controller detect the database server operating system or server errors.	Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server
(2) Arbitration processing	An arbitration command is executed to check the status of the other server's operating system or server.	arbitration_command_timeout in the server configuration file of the database server
(3) Fencing	If the operating system or server status of the other server is abnomal in (2), it fences the other server and isolates it from the cluster system. If the operating system or server status of the other server is nomal in (2), this process and (4) are not executed.	fencing_command_timeout in the server configuration file of the database server
(4) Automatic degradation	The automatic degradation is performed. If fencing fails in (3), this procedure is not performed.	-

-: No associated parameters

Figure 2.3 When the Mirroring Controller on the primary server detects an operating system or server error

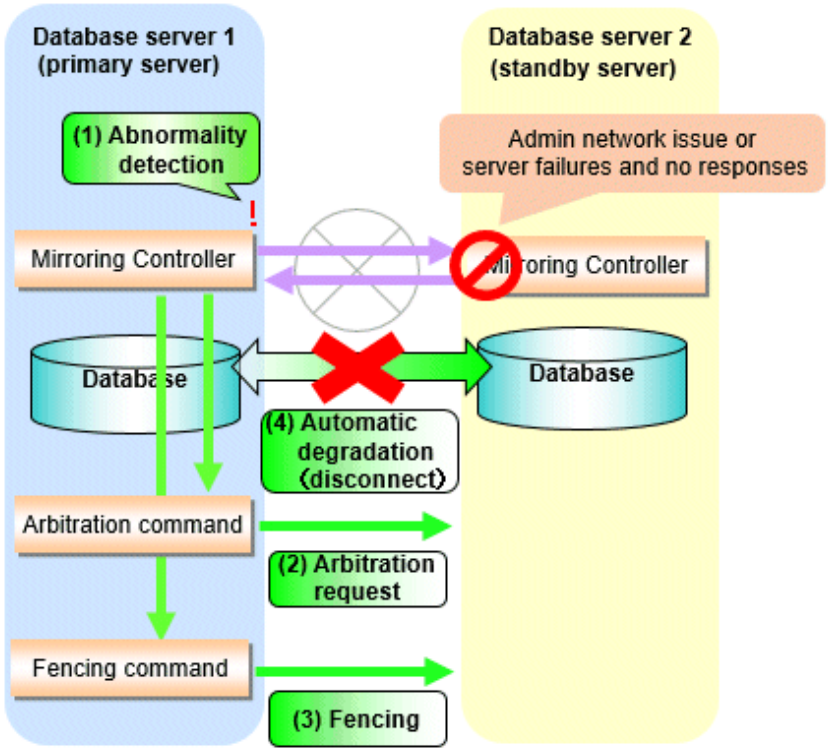
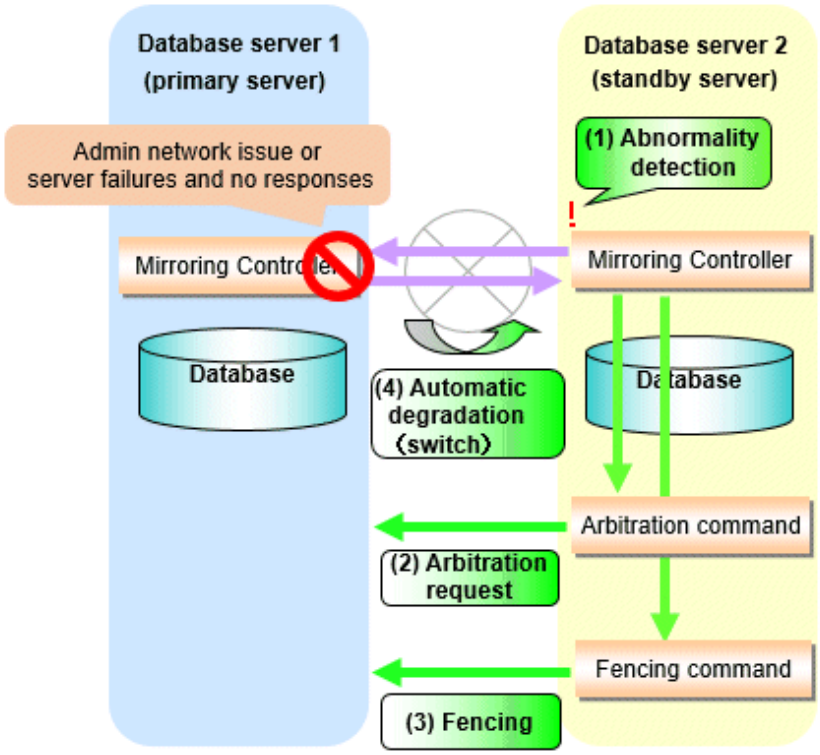


Figure 2.4 When the Mirroring Controller on the standby server detects an operating system or server error



2.11.4.1.3 Tuning Abnormality Monitoring for Operations that Notify Messages

In an operation that notify messages, you can optimize the abnormality detection time by editing the operating system or server abnormality monitoring parameters in the server configuration file of the database server. Refer to "Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server" for information on the operating system or server abnormality monitoring parameters in the server configuration file of the database server. In addition, when the Mirroring Controller detects an error, it does not perform the arbitration processing, fencing, or automatic degradation, but only notification messages is performed.

2.11.4.1.4 Tuning Abnormality Monitoring for Operations that Perform Automatic Degenerate Unconditionally due to Heartbeat Abnormality

In an operation that perform automatic degenerate unconditionally due to heartbeat abnormality, you can optimize the time from operating system or server abnormality detection to automatic degradation by editing the operating system or server abnormality monitoring parameters in the server configuration file of the database server. Refer to "Parameters for the abnormality monitoring of the operating system or server in the server configuration file of the database server" for information on the operating system or server abnormality monitoring parameters in the server configuration file of the database server. In addition, when the Mirroring Controller detects an error, it does not perform the arbitration processing, fencing, or automatic degradation, but only automatic degenerate unconditionally is performed.

Note

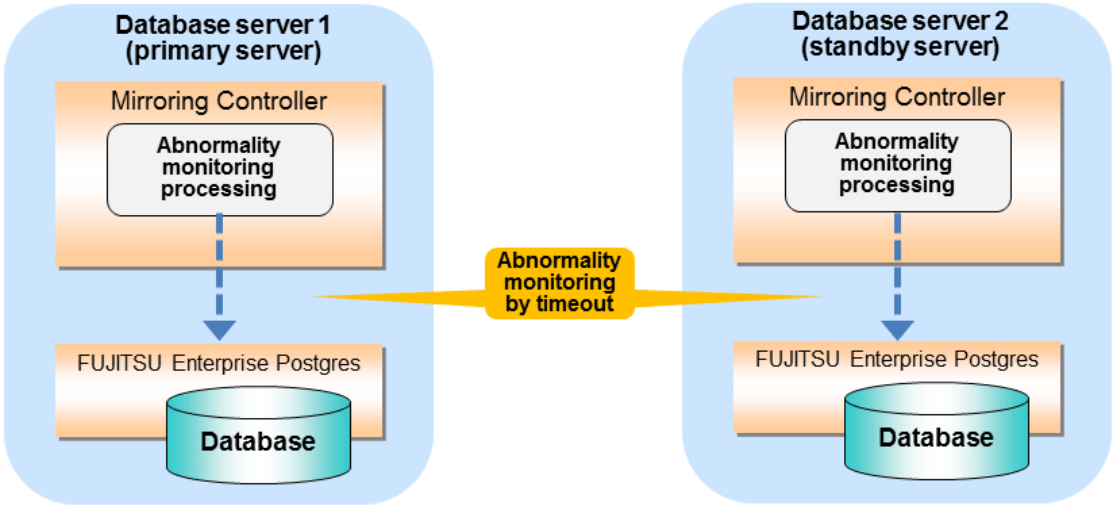
Refer to "Appendix D Notes on Performing Automatic Degradation Immediately after a Heartbeat Abnormality" for notes on the operation that perform automatic degenerate unconditionally due to heartbeat abnormality.

2.11.4.2 Tuning for Abnormality Monitoring of Darabase Processes

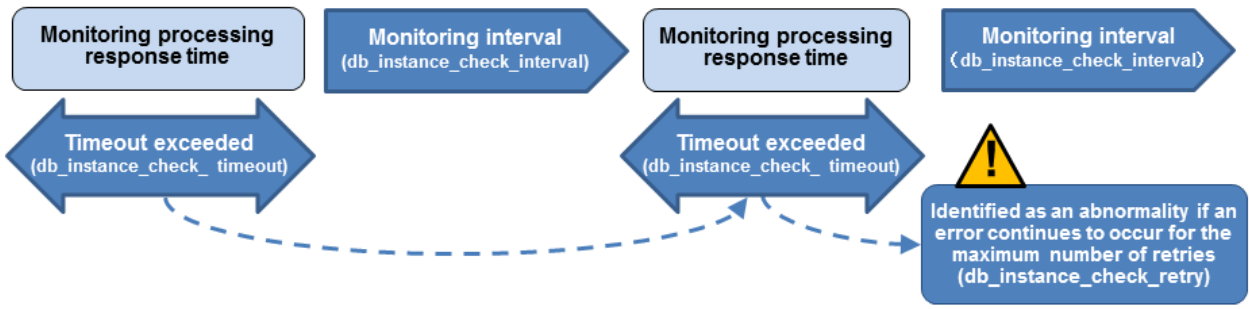
In an abnormality monitoring of database pocesses, you can optimize by editing the following parameters in the server configuration file of the database server.

Table 2.14 Parameters for abnormality monitoring of database processes

Parameter	Description
Abnormality monitoring interval (db_instance_check_interval)	Abnormality monitoring by Mirroring Controller is set so as not to place load on the system, but normally it does not need to be set. (The default is the value set in heartbeat_interval.) (milliseconds)
Timeout for abnormality monitoring of database processes (db_instance_check_timeout)	Take into account the time during which a load is placed continuously on the database. For example, it is envisaged that this parameter will be used in situations such as when performing high-load batch jobs or when a large number of online jobs occur continuously and concurrently. (The default is the value set in heartbeat_timeout.) (seconds)
Abnormality monitoring retries (db_instance_check_retry)	This parameter can be set when needing a safety value for situations in which the value specified for db_instance_check_timeout is exceeded, for example, when using systems with fluctuating loads, however, this parameter does not normally need to be set. (The default is the value set in heartbeat_retry.) (number of times)



Flow of abnormality monitoring by timeout



The expression for calculating the time required to detect an abnormality is shown below.

```
Abnormality detection time = ( db_instance_check_timeout(seconds) +
db_instance_check_interval(milliseconds) / 1000 ) x ( db_instance_check_retry(number of times) + 1 )
```

The abnormality detection time when the default value is used is shown below.

```
Abnormality detection time = ( 1 + 800 / 1000 ) x ( 2 + 1 )
             = 5.4(seconds)
```

Note

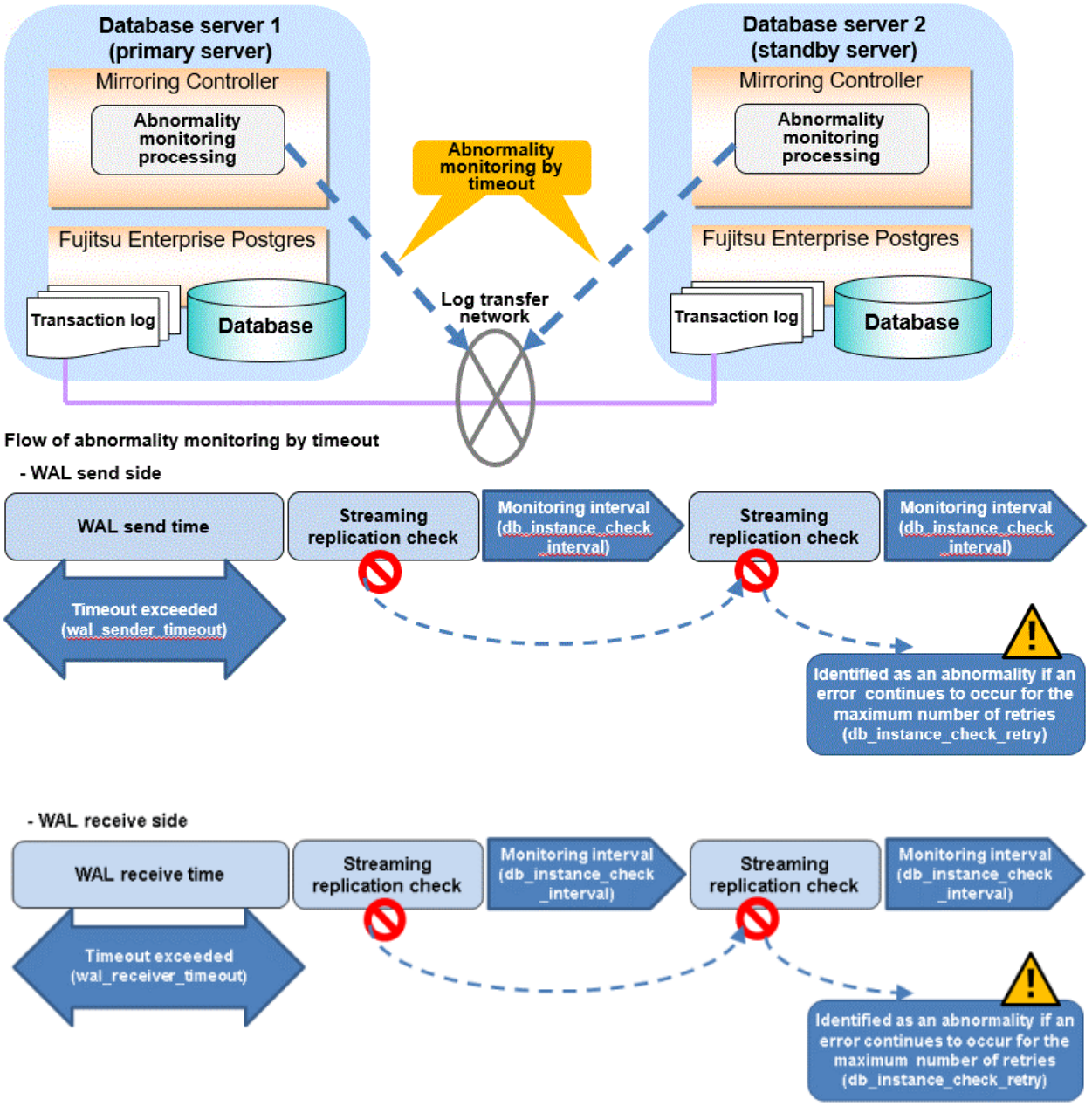
- If the db_instance_timeout_action parameter in *serverIdentifier*.conf is set to "message", and the db_instance_check_timeout parameter is set to a short value, a crash of the database process will be detected as "no response", and it may take time for automatic degradation to occur. Therefore, specify an appropriate timeout for db_instance_check_timeout.
- If a high load on the database and an event that prevents connection to an instance occur at the same time, it is judged as abnormal without retrying monitoring.

2.11.4.3 Tuning for Abnormality Monitoring of Streaming Replication

In an abnormality monitoring of streaming replication, you can optimize by editing the following parameters in the server configuration file of the database server.

Table 2.15 Parameters for abnormality monitoring of streaming replication

Parameter	Description
Abnormality monitoring interval (db_instance_check_interval)	Abnormality monitoring by Mirroring Controller is set so as not to place load on the system, but normally it does not need to be set. (The default is the value set in heartbeat_interval.) (milliseconds)
Abnormality monitoring retries (db_instance_check_retry)	This parameter can be set when needing a safety value, such as when it is anticipated that a temporary log transfer LAN error may occur, but it does not normally need to be set. (The default is the value set in heartbeat_retry.) (number of times)
Timeout for abnormality monitoring of streaming replication (wal_sender_timeout and wal_receiver_timeout in postgresql.conf)	Take into account the capacity and load of the log transfer network and the time during which a load is placed continuously on the database. For example, if there is a succession of data update jobs that generate a high WAL volume, you must configure the settings to avoid misdetection. (The default is 60 seconds.)



The expression for calculating the time required to detect an abnormality is shown below.

```
Abnormality detection time = ( wal_sender_timeout(seconds) +
db_instance_check_interval(milliseconds) / 1000 x ( disk_check_retry(number of times) + 1 ) )  Or,
            = ( wal_receiver_timeout(seconds) + db_instance_check_interval(milliseconds) / 1000 x
( disk_check_retry(number of times) + 1 ) )
```

The abnormality detection time when the default value is used is shown below.

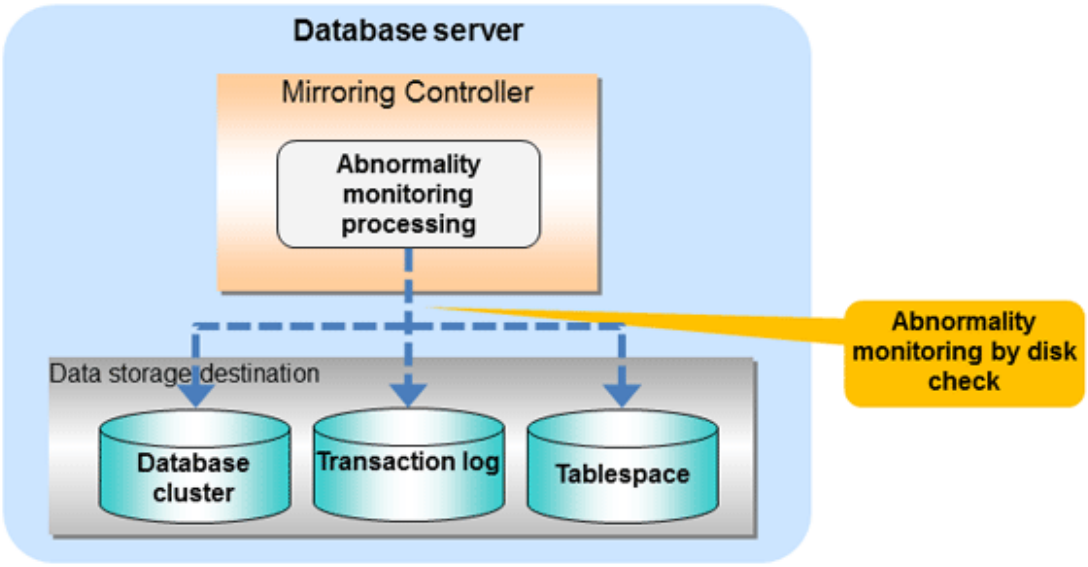
```
Abnormality detection time = 60 + (800 / 1000 x (2 + 1))
            = 62.4(seconds)
```

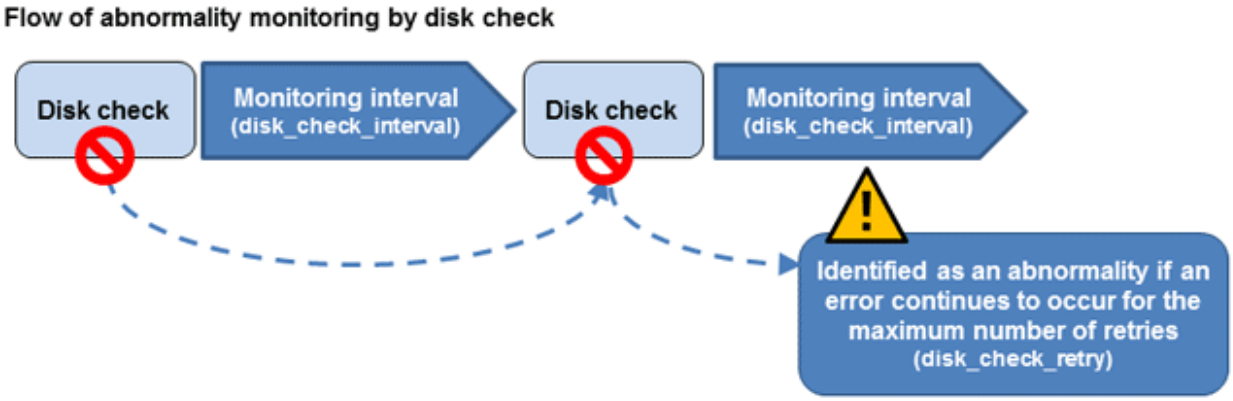
2.11.4.4 Tuning for Disk Abnormality Monitoring

In an abnormality monitoring of the disk, you can optimize by editing the following parameters in the server configuration file of the database server.

Table 2.16 Parameters for disk abnormality monitoring

Parameter	Description
Abnormality monitoring interval (disk_check_interval)	Abnormality monitoring by Mirroring Controller is set so as not to place load on the system, but normally it does not need to be set. (The default is the value set in heartbeat_interval.) (milliseconds)
Abnormality monitoring retries (disk_check_retry)	This parameter can be set when needing a safety value, such as when it is anticipated that a temporary disk input/output error may occur, but normally it does not need to be set. (The default is the value set in heartbeat_retry.) (number of times)





The expression for calculating the time required to detect an abnormality is shown below.

```
Abnormality detection time = disk_check_interval (milliseconds) / 1000 x ( disk_check_retry(number of times) + 1 )
```

The abnormality detection time when the default value is used is shown below.

```
Abnormality detection time = 800 / 1000  x ( 2 + 1 )
            = 2.4(seconds)
```

Note

- The tuning described above impacts on the time taken from detection of a timeout until switching the primary server. Therefore, modify the values while taking into account the switch/disconnection time, using a design for which misdetection does not occur.
- Immediately selecting automatic degradation when a heartbeat abnormality occurs in operating system or server heartbeat monitoring risks causing split brain. Refer to "Appendix D Notes on Performing Automatic Degradation Immediately after a Heartbeat Abnormality" for details.

Information

Mirroring Controller uses connections to database instances and SQL access to monitor abnormality in some resources targeted for monitoring. The connection destination database names and connection user names used for abnormality monitoring conform to the parameters in the server configuration file. The application name is "mc_agent".

2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances

Multiplexed instances and Mirroring Controller can be started and stopped automatically in line with the starting and stopping of the operating system of the database server.

However, to stop the operating system on the primary server, it is necessary to stop the operating system on the standby server so that no unnecessary automatic switching or errors occur.

Note

To guarantee the startup sequence of Mirroring Controller on the primary and standby servers, first confirm that the primary server has started, and then start the standby servers in sequence.

The startup sequence of the Mirroring Controller process on the database server and the Mirroring Controller arbitration process on the arbitration server is not guaranteed. If the arbitration server cannot be started first, execute the mc_ctl command in start mode with the --async-connect-arbiter option specified to start the Mirroring Controller process.

You can configure the Windows service to perform automatic start and stop of Mirroring Controller.

Setting automatic start and stop of a multiplexed instance

No settings are required for Mirroring Controller to start and stop an instance.

Note

Do not configure the Windows service of a multiplexed instance to perform automatic start.

Configuring automatic start and stop for database multiplexing mode

Configuring during setup

When registering Mirroring Controller to the Windows service in "2.4.1 Setting Up Database Multiplexing Mode on the Primary Server" and "2.5.1 Setting Up Database Multiplexing Mode on the Standby Server", specify "auto" for the -S option of the register mode used with the mc_ctl command.

```
Example)
```

```
> mc_ctl register -M D:\mcdir\inst1 -P ******** -S auto
```

Changing the configuration after setup

Use the sc config command to change the configuration of the Windows service of Mirroring Controller.

Example)
The following is an example using the registered service name "Mirroring_Controller_inst1".

```
> sc config "Mirroring_Controller_inst1" start= auto
```

See

Refer to documentation such as Windows Help and Support for the sc command for information on how to configure the service.

Information

You can check the registration status in the Windows service window or by using the sc qc command.

2.13 Setting Automatic Start and Stop of the Mirroring Controller Arbitration Process

You can automatically start or stop the Mirroring Controller arbitration process when the operating system on the arbitration server is started or stopped.

Linux

Note

If you start the Mirroring Controller arbitration process, wait for time correction, network setup, and so on.

Perform the following procedure:

1. Create a unit file.

   Copy the unit file sample stored in the directory below, and revise it to match the target instance.

   Sample file

   ```
   /installDir/share/mcarboi.service.sample
   ```

   Example)
   In the following example, the installation directory is "/opt/fsepv<x>assistant", and the identifier of the arbitration process is "arbiter1". Note that "<x>" indicates the product version.

   ```
   # cp /opt/fsepv<x>assistant/share/mcarboi.service.sample /usr/lib/systemd/system/
   mcarboi_arbiter1.service
   ```

   Revise the underlined portions of the options below in the unit file.

Section	Option	Specified value	Description
Unit	Description	Fujitsu Enterprise Postgres Mirroring Controller Arbiter <*arbitrationProcessId*>	Specifies the feature overview. Specifies the identifier of the targeted arbitration process. (*1)
Service	ExecStart	/bin/bash -c '*installDir*/bin/mc_arb_std start *installDir mirroringControllerArbitrationProcessMgmtDir mc_arbOption*'	Command to be executed when the service is started.

Section	Option	Specified value	Description
			Specify the option you want to add when the mc_arb command is executed without the -M option in the mc_arb option. Note that the content specified in this mc_arb option is carried over from the mc_arb_std command in "Specified value" to the mc_arb command.
	ExecStop	/bin/bash -c '*installDir*/bin/ mc_arb_std stop *installDir mirroringControllerArbitrationProcessMgmtDir mc_arbOption*'	Command to be executed when the service is stopped. Specify the option you want to add when the mc_arb command is executed without the -M option in the mc_arb option. Note that the content specified in this mc_arb option is carried over from the mc_arb_std command in "Specified value" to the mc_arb command.
	User	User	Specify the account of the operating system user.
	Group	Group	Specify the group to which the user belongs.

*1: The arbitration process identifier used here is a name for identifying the Mirroring Controller arbitration process.

The naming conventions for identifying the Mirroring Controller arbitration process are as follows:

- Up to 16 bytes
- The first character must be an ASCII alphabetic character
- The other characters must be ASCII alphanumeric characters

2. Enable automatic start and stop.

   As the operating system superuser, use the systemctl command to enable automatic start and stop.

   Example)

```
# systemctl enable mcarboi_arbiter1.service
```

Windows

You can configure the Windows service to perform automatic start and stop.

Configuring during setup

When registering the Mirroring Controller arbitration process as a Windows service in "2.3.1 Configuring the Arbitration Server", specify "auto" for the -S option of the register mode used with the mc_arb command.

Example)

```
> mc_arb register -M D:\mcdir\inst1 -P ******** -S auto
```

Changing the configuration after setup

Use the sc config command to change the configuration of the Windows service of the Mirroring Controller arbitration process.

Example)
The configuration of the registered service name "Mirroring_Controller_Arbiter1" is changed.

```
> sc config "Mirroring_Controller_Arbiter1" start= auto
```

See

Refer to sc command help for information on how to configure the service.

Information

You can check the registration status in the Windows service window or by using the sc qc command.

2.14 Backup Operation

This section explains the backup operation for database multiplexing mode.

2.14.1 Backing up Database Multiplexing Mode Information

When changing the Mirroring Controller settings, in addition to backing up the database, back up the configuration file in the Mirroring Controller management directory so that the Mirroring Controller settings are not lost.

When the arbitration server is used for automatic degradation, also back up the configuration file in the Mirroring Controller arbitration process management directory.

2.14.2 Database Backup Operation

Using database multiplexing mode is the same as obtaining the backup data on the standby server as a safeguard against a disk failure. Note that all server disks may be corrupted due to some cause.

As a safeguard against this type of case, execute the pgx_dmpall command on the primary server to create the backup data.

However, it is not definite as to which server runs as the primary server, so ensure that the pgx_dmpall command is executed periodically on all servers, so that the backup data will be obtained. For example, create a script to obtain the backup data, and set it in the operation management software.

Point

When the pgx_dmpall command is executed on the standby server, it will not match the statuses, however the error message shown below will be output and return the value "1".

If a script that ignores only this type of error is executed on all servers, the backup data of the primary server can be obtained.

Error message

```
ERROR:recovery is in progress (10095)
```

Note

- Consider the possibility that the server that runs as the primary server may be destroyed alongside the backup data, so it is recommended to promote another server to become the primary server, and then back up the data on the new primary server without waiting for the next scheduled backup.
- Specify the same backup directory name for the primary and standby servers. If different backup directory names are specified, and recovery is performed using the backup data of the other server, the recovery cannot be performed correctly.

See

- Period backups allow shorter recovery time and reduction in disk usage. Refer to "Backing Up the Database" in the Operation Guide for details on the backup operation.
- Refer to "Chapter 4 Action Required when an Error Occurs in Database Multiplexing Mode" for details on recovery based on the backup data that was obtained using the pgx_dmpall command.

Chapter 3 Operations in Database Multiplexing Mode

This chapter describes the periodic operations that are performed when running database multiplexing mode.

The periodic operations are the same as the operations on a single server.

See

Refer to "Periodic Operations" in the Operation Guide for information on the periodic operations.

3.1 Starting and Stopping the Mirroring Controller Arbitration Process

This section describes how to start and stop the Mirroring Controller arbitration process.

3.1.1 Starting the Mirroring Controller Arbitration Process

Linux

While the Mirroring Controller arbitration process is in a stopped state, execute the mc_arb command in start mode to start the Mirroring Controller arbitration process.

Example)

```
$ mc_arb start -M /mcarb_dir/arbiter1
```

Windows

The Mirroring Controller arbitration process can be started using one of the following options:

- Using the mc_arb command
- Starting the service on system startup

Using the mc_arb command

While the Mirroring Controller arbitration process is in a stopped state, execute the mc_arb command from the command prompt to start the Mirroring Controller arbitration process.

Example)

```
> mc_arb start -M D:\mcarb_dir\arbiter1
```

Starting the service on system startup

Specify automatic start when registering the Mirroring Controller arbitration process to the Windows service during setup of database multiplexing mode. Accordingly, the Mirroring Controller arbitration process service will start on startup of the operating system.

See

Refer to the Reference for information on how to specify the mc_arb command.

3.1.2 Stopping the Mirroring Controller Arbitration Process

Linux

While the Mirroring Controller arbitration process is running, execute the mc_arb command in stop mode to stop the Mirroring Controller arbitration process.

```
Example)

$ mc_arb stop -M /mcarb_dir/arbiter1
```

Windows

The Mirroring Controller arbitration process can be stopped using one of the following options:

- Using the mc_arb command
- Stopping the service

Using the mc_arb command

While the Mirroring Controller arbitration process is running, execute the mc_arb command in stop mode from the command prompt to stop the Mirroring Controller arbitration process.

```
Example)

> mc_arb stop -M D:\mcarb_dir\arbiter1
```

Stopping the service

Select [Administrative Tools], then [Services] to open the [Services] window, and then select the Mirroring Controller service and click the [Stop] menu.

See

Refer to the Reference for information on how to specify the mc_arb command.

Note

- The arbitration server will be forcibly stopped when the service is stopped.
- Before shutting down the operating system on the arbitration server, either stop the Mirroring Controller on the primary server or standby server or shut down the operating system on the primary server or standby server.

3.2 Starting and Stopping Mirroring Controller

This section describes how to start and stop the Mirroring Controller.

3.2.1 Starting Mirroring Controller

Mirroring Controller can be started using one of the following:

- Using the mc_ctl command
- Starting a service on system startup

Note

Mirroring Controller must be started by a user with administrator privileges (user ID belonging to the Administrators group).

Using the mc_ctl command

While Mirroring Controller is in a stopped state, execute the mc_ctl command from the command prompt to start the instance and Mirroring Controller.

Enabling automatic switch/disconnection

Execute the mc_ctl command in start mode.

Example)

```
> mc_ctl start -M D:\mcdir\inst1
```

Disabling automatic switch/disconnection

Execute the mc_ctl command in start mode with the -F option specified.

Example)

```
> mc_ctl start -M D:\mcdir\inst1 -F
```

Point

- To start the Mirroring Controller process only, execute the mc_ctl command in start mode with the --mc-only option specified.
- After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.
- When the arbitration server is used for automatic degradation, the Mirroring Controller process startup fails on the database server if the Mirroring Controller arbitration process has not been started on the arbitration server in advance. However, even if the Mirroring Controller arbitration process cannot be started in advance, the Mirroring Controller process can be started by specifying the --async-connect-arbiter option in the mc_ctl command.

See

Refer to the Command Reference for information on how to specify the mc_ctl command.

Starting a service on system startup

Specify automatic start when registering Mirroring Controller to the Windows service during setup of database multiplexing mode. Accordingly, the Mirroring Controller service will start on startup of the operating system.

See

Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for details.

Note

- Mirroring Controller startup usually fails if the standby server is mistakenly started as the primary server or if the old primary server is not recovered after the switch and is then mistakenly started as the primary server. However, if the admin network is disconnected, then startup does not fail, and both servers may become primary servers. Therefore, ensure that the admin network is connected before starting Mirroring Controller.
- When only the instance is started without starting Mirroring Controller, the impact will be as follows:
  - Enabling automatic switch/disconnection

    Features such as automatic switch and automatic disconnection will not work until Mirroring Controller is started.

- Disabling automatic switch/disconnection

  Errors indicated in "1.1 What is Database Multiplexing Mode" will not be detected until Mirroring Controller is started.

3.2.2 Stopping Mirroring Controller

Mirroring Controller can be stopped using one of the following ways:

- Using the mc_ctl command
- Stopping the service

Note

Mirroring Controller must be stopped by a user with administrator privileges (user ID belonging to the Administrators group).

Using the mc_ctl command

While Mirroring Controller is running, execute the mc_ctl command from the command prompt in stop mode to stop Mirroring Controller.

```
Example)
```

```
> mc_ctl stop -M D:\mcdir\inst1
```

Point

To stop the Mirroring Controller process only, execute the mc_ctl command in stop mode with the --mc-only option specified.

See

Refer to the Command Reference for information on how to specify the mc_ctl command.

Stopping the service

Select [Administrative Tools], then [Services] to open the [Services] window, and then select the Mirroring Controller service and click the [Stop] menu.

Note

- To prevent an unintended automatic switch, before shutting down the operating system on the primary server, you must stop the Mirroring Controller, or shut down the operating system on the standby server.
- If you stop the Mirroring Controller by stopping the service, you must firstly exit all applications or programs that are using the instance that is to be stopped.
- When only the instance is stopped without stopping Mirroring Controller, the impact will be as follows:
  - Enabling automatic switch/disconnection

    Mirroring Controller determines that an error has occurred in the instance, and performs an unnecessary automatic switch. Automatic switch may also stop working correctly in some cases.
  - Disabling automatic switch/disconnection

    Mirroring Controller determines that an error has occurred in the instance, and outputs an error to the event log.

3.3 Checking the Database Multiplexing Mode Status

3.3.1 Checking the Status of the Database Server

This section describes how to check the status of the database server.

Check the multiplexed database status by executing the mc_ctl command in status mode.

Additionally, errors can be detected by monitoring the Mirroring Controller messages. If the status or messages are monitored periodically, you can react quickly following an automatic switch failure.

Checking the status of the multiplexing database

When the mc_ctl command is executed, the details of the multiplexing configuration, information about whether switch is possible following the error, and location and details of the error that caused the switch or disconnection are displayed.

After starting database multiplexing mode, execute the mc_ctl command in status mode to check the multiplexing status.

An example of the status displayed when the mc_ctl command is executed is shown below.

Example)

```
> mc_ctl status -M D:\mcdir\inst1

mirroring status
----------------
switchable
server_id  host_role       host            host_status  db_proc_status       disk_status
-------------------------------------------------------------------------------------------------
----
server1    primary         192.0.2.100     normal       normal                 normal
server2    standby         192.0.2.110     normal       normal                 normal
```

Checking the status of connection to the Mirroring Controller arbitration process

When the arbitration server is used for automatic degradation, the status of the connection to the Mirroring Controller arbitration process can be checked by specifying the --arbiter option. If the output status is "online", it indicates that an arbitration request can be made from the database server to the arbitration server. When the arbitration server is used for automatic degradation, regularly execute the command in status mode with the --arbiter option specified and check that the output status is "online".

Example)
The mc_ctl command is executed with the --arbiter option specified, and the status is output.

```
> mc_ctl status --arbiter -M D:\mcdir\inst1

arbiter_id   host            status
-------------------------------------
arbiter      192.0.3.120     online
```

Checking the status of data synchronization

Additionally, by referencing the pg_stat_replication statistics view on the primary server, the data synchronization status can be confirmed. However, when creating the monitoring program, note that the content of pg_stat_replication may be changed in the future.

The following example shows that the locations of the transaction log after it is sent and received (sent_lsn, replay_lsn) match, and that they are fully synchronized.

Example)

```
postgres=# select * from pg_stat_replication;
-[ RECORD 1 ]----+------------------------------
pid              | 10651
usesysid         | 10
usename          | fsep
```

```
application_name | standby
client_addr      | 192.0.2.210
client_hostname  |
client_port      | 55098
backend_start    | 2022-03-23 11:17:49.628793+09
backend_xmin     |
state            | streaming
sent_lsn         | 0/3000060
write_lsn        | 0/3000060
flush_lsn        | 0/3000060
replay_lsn       | 0/3000060
write_lag        |
flush_lag        |
replay_lag       |
sync_priority    | 1
sync_state       | sync
reply_time       | 2022-03-23 11:23:27.703366+09
```

See

- Refer to "mc_ctl" in Reference for information on the command.
- Refer to "Notes on Application Compatibility" in the Application Development Guide for information on retaining application compatibility.
- Refer to "The Statistics Collector" in "Server Administration" in the PostgreSQL Documentation for details on pg_stat_replication.

3.3.2 Checking the Status of the Arbitration Server

This section describes how to check the status of the arbitration server.

The status of the connection between the Mirroring Controller arbitration process and primary server/standby server can be checked by executing the mc_arb command in status mode.

The example below executes the mc_arb command, and shows the status.

Linux

Example)

```
$ mc_arb status -M /mcarb_dir/arbiter1

server_id    host            status
-------------------------------------
server1      192.0.3.100     online
server2      192.0.3.110     online
```

Windows

Example)

```
> mc_arb status -M D:\mcarb_dir\arbiter1

server_id    host            status
-------------------------------------
server1      192.0.3.100     online
server2      192.0.3.110     online
```

3.4 Manually Switching the Primary Server

The primary server cannot be switched automatically in the following case:

- If automatic switch/disconnection is disabled
- If output of messages is selected for heartbeat abnormalities during heartbeat monitoring of the operating system or server and the operating system/server crashes or becomes unresponsive

In this case, to manually switch the primary server, execute the mc_ctl command in switch mode on either the primary server or the standby server.

Example)

```
> mc_ctl switch -M D:\mcdir\inst1
```

Point

If automatic switch/disconnection is enabled, it is possible to perform switch of primary server at any time.

3.5 Manually Disconnecting the Standby Server

The procedure to perform disconnection of the standby server differs depending on whether the automatic switch/disconnection is enabled or disabled.

If automatic switch/disconnection is enabled

Execute the mc_ctl command in stop mode on the standby server.

Example)

```
> mc_ctl stop -M D:\mcdir\inst1
```

If automatic switch/disconnection is disabled

1. Execute the mc_ctl command in stop mode on the standby server.

   Example)

   ```
   > mc_ctl stop -M D:\mcdir\inst1
   ```

2. Comment out the synchronous_standby_names parameter in the postgresql.conf file on the primary server.
3. Execute the pg_ctl command in reload mode on the primary server.

   Example)

   ```
   > pg_ctl reload -D D:\database\inst1
   ```

3.6 Action Required when a Heartbeat Abnormality is Detected

The message below is output when a heartbeat abnormality is detected during heartbeat monitoring of operating systems or servers:

```
detected an error on the monitored object "server(server identifier name)": no response:ping timeout
(MCA00019)
```

If the heartbeat_error_action parameter in *serverIdentifier*.conf is set to "message", even if automatic switch/disconnection is enabled and Mirroring Controller is started, automatic switch/disconnection is not performed when a heartbeat abnormality is detected. Therefore, user action will be necessary.

This section explains the action required when the heartbeat_error_action parameter is set to "message" and a heartbeat abnormality is detected.

1. Identify the cause of the heartbeat abnormality. The possible causes are below:
   - The remote operating system or server crashed or is unresponsive

- An admin network issue occurred

2. Address the cause identified in step 1.

   - The remote operating system or server crashed or is unresponsive

     Manually perform switch or disconnection using the mc_ctl command.

   - An admin network issue occurred

     Refer to "Chapter 4 Action Required when an Error Occurs in Database Multiplexing Mode", and recover the database multiplexing system.

3.7 Monitoring Mirroring Controller Messages

The messages that are output by Mirroring Controller are output to both the database server and the arbitration server. If the automatic switch fails, for example, an important message related to the continuation of the operation may be output, so ensure that the system log messages are monitored.

If the arbitration server is used for automatic degradation, monitor messages on both the database server and the arbitration server.

Message output destination on the database server

Messages are output to the event log.

Message output destination on the arbitration server

Linux

Messages are output to the system log.

Windows

Messages are output to the event log.

Point

- To monitor message types considered to be important, an operating system setting must be configured beforehand. Refer to the operating system manuals, check if the message is of a message type that is monitored to be output to the system log, and configure the setting if required.
- If the heartbeat_error_action parameter in *serverIdentifier*.conf is set to "message", only message output is performed when a heartbeat abnormality is detected during heartbeat monitoring of operating systems and servers - automatic switch/disconnection is not performed. Therefore users need to monitor the messages. Refer to "3.6 Action Required when a Heartbeat Abnormality is Detected" for details.

Display format on the database server

```
eventSourceName[processId]: messageType: messageContent (messageNumber)
```

Specify the event source name in the event_source parameter of the serverIdentifier.*conf* file of the database server.

The message types output by Mirroring Controller, their severity, and their corresponding value in the event log are shown in the table below.

Table 3.1 Message type, severity, and corresponding value in the event log

Message type	Severity	Meaning	Event log
INFO	Information	Provides information that does not fall under LOG or NOTICE.	INFORMATION
LOG		Provides information recognized as a particularly important event in tracing the operation history. (Example: Automatic switch is complete)	

Message type	Severity	Meaning	Event log
NOTICE	Notice	Outputs information that takes into account the user instructions within the program in response to an executed or automatically executed process.	
WARNING	Warning	Provides a warning, for example it will soon be impossible to maintain multiplexing capabilities.	WARNING
ERROR	Error	Reports that an error other than FATAL or PANIC has occurred.	ERROR
FATAL		Reports that an abnormality was detected in the multiplexed database systems requiring recovery of the system, and also the content and cause of the abnormality.	
PANIC		Reports that an abnormality was detected in all multiplexed database systems requiring immediate recovery of the system, and also the content and cause of the abnormality.	

The message severity has the following meanings:

- Information

  Informational status. A message that was reported by the system is displayed. No action is required.

- Notice

  Informational status, but a message that should be noted is displayed. If necessary, take the actions described in the "Action" section of the message.

- Warning

  No error has occurred, but the user is requested to check, and take action. Take the actions described in the "Action" section of the message.

- Error

  An error has occurred. Take the actions described in the "Action" section of the message.

Display format on the arbitration server

Linux

```
programName[processId]: messageType: messageText (messageNumber)
```

Specify the program name in the syslog_ident parameter of the arbitration.conf file of the arbitration server.

Windows

```
eventSourceName[processId]: messageType: messageText (messageNumber)
```

Specify the event source name in the event_source parameter of the arbitration.conf file of the arbitration server.

The message types output by Mirroring Controller, their severity, and their corresponding value in the output destination log are shown in the table below.

Table 3.2 Message type, severity, and corresponding value in the output destination log

Message type	Severity	Meaning	System log (Linux)	Event log (Windows)
INFO	Information	Provides information not categorized as LOG or NOTICE.	INFO	INFORMATION
LOG		Provides information recognized as a particularly important event in tracing the operation history. (Example: Automatic switch is complete)		

Message type	Severity	Meaning	System log (Linux)	Event log (Windows)
NOTICE	Notice	Outputs information that takes into account the user instructions within the program in response to an executed or automatically executed process.	NOTICE	
WARNING	Warning	Provides a warning, for example it will soon be impossible to perform the arbitration process.	WARNING	WARNING
ERROR	Error	Reports that an error other than FATAL or PANIC has occurred.	ERROR	ERROR
FATAL		Reports that an abnormality was detected in the arbitration server requiring recovery of the system, and also the content and cause of the abnormality.	CRIT	
PANIC		Reports that an abnormality was detected in the arbitration server requiring immediate recovery of the system, and also the content and cause of the abnormality.	ALERT	

The message severity has the following meanings:

- Information

  Informational status. A message that was reported by the system is displayed. No action is required.

- Notice

  Informational status, but a message that should be noted is displayed. If necessary, take the actions described in the "Action" section of the message.

- Warning

  No error has occurred, but the user is requested to check, and take action. Take the actions described in the "Action" section of the message.

- Error

  An error has occurred. Take the actions described in the "Action" section of the message.

3.8 Server Maintenance

To perform maintenance tasks such as periodic server inspections and the application of updates for software products including the operating system, you must perform a planned stop of the server, and then perform the maintenance.

3.8.1 Rolling Updates

In database multiplexing mode, rolling updates, that perform the maintenance for the servers that comprise the cluster system, can be performed while jobs continue.

First, perform the maintenance for the standby server, and then switch the standby server to the primary server. Then, perform the maintenance for the original primary server that was switched to the standby server. This enables maintenance to be performed while jobs continue.

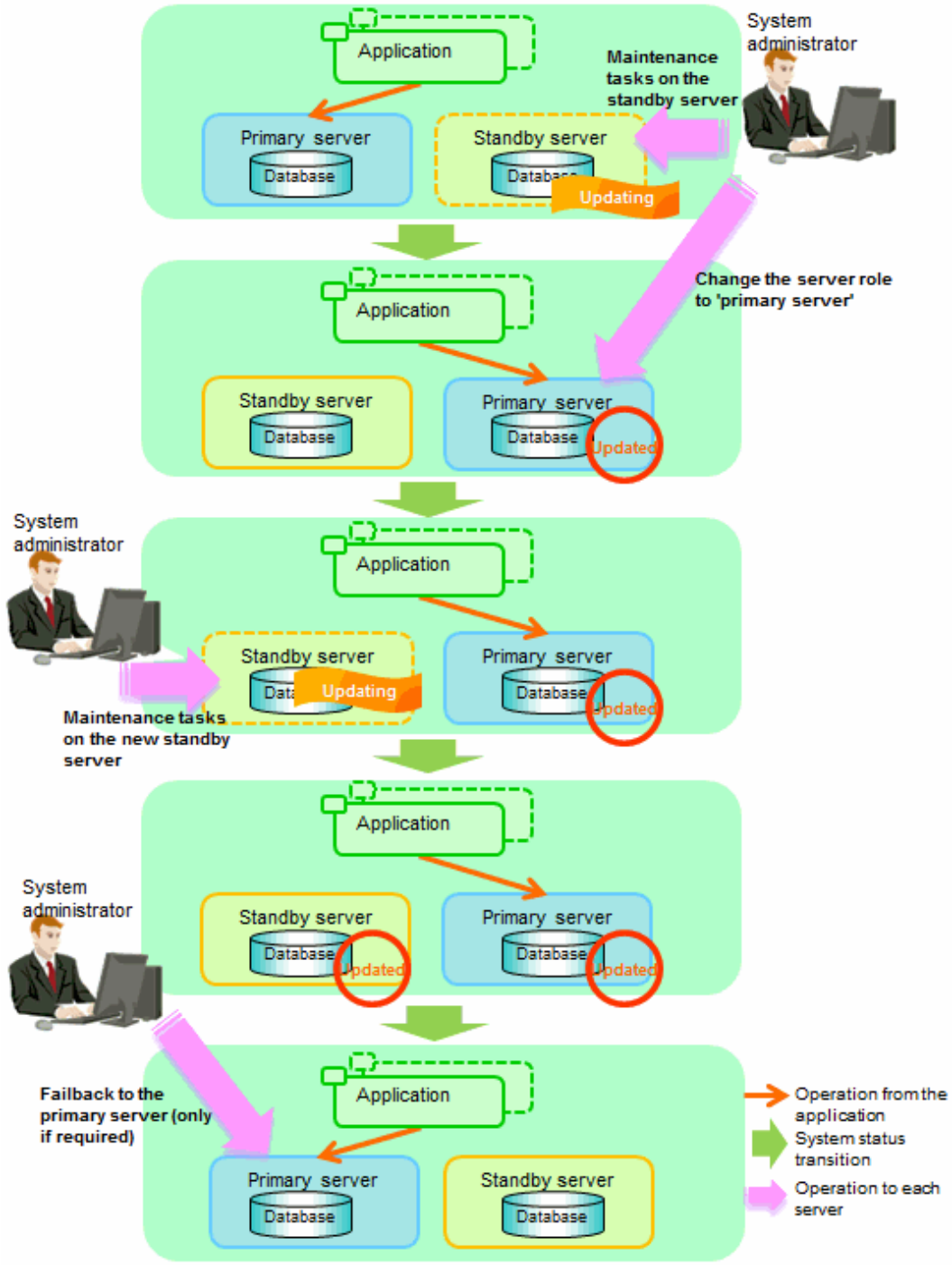
Note that arbitration server maintenance can be performed without affecting database server operation, so it is not necessary to consider rolling update.

See

If the downtime due to the maintenance of the standby server is expected to be long, refer to "Standby server downtime" in "3.9.1 Changes Required when the Standby Server is Stopped".

The flow of a rolling update is shown below.

Figure 3.1 Performing a Rolling Update



Perform the following procedure as shown in the above figure:

Standby server maintenance tasks

1. To perform the maintenance on the standby server, stop Mirroring Controller.

   Example)

   ```
   > mc_ctl stop -M D:\mcdir\inst1
   ```

2. Ensure that Mirroring Controller has completely stopped.

   If the multiplexed instances and Mirroring Controller have been configured on the standby server to start and stop automatically when the operating system of the database server is started or stopped, cancel the setting to start and stop automatically.

   ### See

   Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for information on how to configure the multiplexed instances and Mirroring Controller to start and stop automatically when the operating system of the database server start and stops.

   This task should be performed by the instance administrator user with administrator privileges.

   Use the sc config command to disable automatic start of multiplexed instances and Mirroring Controller from the Windows service.

   ```
   Example)
   The following is an example using the registered service name "Mirroring_Controller_inst1".
   ```

   ```
   > sc config "Mirroring_Controller_inst1" start= demand
   ```

   ### Information

   You can use the sc qc command to check the registration status.
   Refer to documentation such as Windows Help and Support for the sc command for information on registry content.

3. Perform maintenance tasks.

4. Create a copy of the primary server instance on the standby server.

   Execute the pg_basebackup command to create data in the standby server by synchronizing with the primary server.

   ```
   Example)
   ```

   ```
   > pg_basebackup -D D:\database\inst1 -X fetch --waldir=E:\transaction\inst1 --progress --verbose
   -R --dbname="application_name=standbyServerName" -h primaryServerHostName -p
   primaryServerPortNumber
   ```

   ### See

   The procedure for copying the primary server instance to the standby server is the same as the procedure for setting up the standby server.

   Refer to "2.5.2 Creating, Setting, and Registering the Standby Server Instance", and then perform the recovery.

5. Check the settings for automatic start and stop of the multiplexed instances and Mirroring Controller.

   If the multiplexed instances and Mirroring Controller were configured in step 2 to not start and stop automatically when the operating system of the database server starts and stops, then change the settings back. This step can be skipped if automatic start and stop are not required.

   This task should be performed by an instance administrator user with administrator privileges.

   Use the sc config command to enable automatic start of multiplexed instances and Mirroring Controller from the Windows service.

   ```
   Example)
   The following is an example using the registered service name "Mirroring_Controller_inst1".
   ```

   ```
   > sc config "Mirroring_Controller_inst1" start= auto
   ```

Information

You can use the sc qc command to check the registration status.
Refer to documentation such as Windows Help and Support for the sc command for information on registry content.

6. Start (rebuild) Mirroring Controller on the standby server.

   This operation is required when determining the maintenance tasks on the standby server.

   Enabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode.

   Example)

```
> mc_ctl start -M D:\mcdir\inst1
```

   Disabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

   Example)

```
> mc_ctl start -M D:\mcdir\inst1 -F
```

Point

After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

Switching to the primary server

To perform the maintenance on the primary server, execute the mc_ctl command in the switch mode on the primary server or the standby server.

Example)

```
> mc_ctl switch -M D:\mcdir\inst1
```

When the switch is complete, the synchronous_standby_names parameter in the postgresql.conf file of the new primary server will be commented as follows:

Example)

```
#synchronous_standby_names = 'primary'
```

New standby server maintenance tasks

1. Stop the Mirroring Controller.

   On the new standby server (the primary server before the switch), execute the mc_ctl command in stop mode.

   Example)

```
> mc_ctl stop -M D:\mcdir\inst1
```

2. Ensure that Mirroring Controller has completely stopped.

   If the multiplexed instances and Mirroring Controller have been configured on the new standby server to start and stop automatically when the operating system of the database server is started or stopped, cancel the setting to start and stop automatically now.

See

Refer to "2.12 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for information on how to configure the multiplexed instances and Mirroring Controller to start and stop automatically when the operating system of the database server starts and stops.

This task should be performed by an instance administrator user with administrator privileges.

Use the sc config command to disable automatic start of multiplexed instances and Mirroring Controller from the Windows service.

```
Example)
The following is an example using the registered service name "Mirroring_Controller_inst1".
```

```
> sc config "Mirroring_Controller_inst1" start= demand
```

Information

You can use the sc qc command to check the registration status.
Refer to documentation such as Windows Help and Support for the sc command information on registry content.

3. Perform the maintenance on the new standby server that was stopped.

4. Create a copy of the new primary server instance on the new standby server.

   Execute the pg_basebackup command to create data in the new standby server by synchronizing with the new primary server.

   ```
   Example)
   ```

   ```
   > pg_basebackup -D D:\database\inst1 -X fetch --waldir=\transaction\inst1 --progress --verbose -
   R --dbname="application_name=standbyServerName" -h primaryServerHostName -p
   primaryServerPortNumber
   ```

   ### See

   The procedure for copying the primary server instance to the standby server is the same as the procedure for setting up the standby server.

   Refer to "2.5.2 Creating, Setting, and Registering the Standby Server Instance", and then perform the recovery.

5. Check the settings for automatic start and stop of the multiplexed instances and Mirroring Controller.

   If the multiplexed instances and Mirroring Controller were configured in step 2 to not start and stop automatically when the operating system of the database server starts and stops, then change the settings back. This step can be skipped if automatic start and stop are not required.

   This task should be performed by an instance administrator user with administrator privileges.

   Use the sc config command to enable automatic start of multiplexed instances and Mirroring Controller from the Windows service.

   ```
   Example)
   The following is an example using the registered service name "Mirroring_Controller_inst1".
   ```

   ```
   > sc config "Mirroring_Controller_inst1" start= auto
   ```

   ### Information

   You can use the sc qc command to check the registration status.
   Refer to the document such as Windows Help and Support for the sc command for information on registry content.

6. After the maintenance is complete, edit the following parameters in the postgresql.conf file of the standby server as required.

   Copying an instance results in the value of the synchronous_standby_names parameter becoming the specified value on the primary server. Therefore, correct it to the specified value on the standby server. If the parameter was commented out, then you must uncomment it.

7. On the standby server, start (rebuild) Mirroring Controller.

   Enabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode.

   Example)

```
> mc_ctl start -M D:\mcdir\inst1
```

   Disabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

   Example)

```
> mc_ctl start -M D:\mcdir\inst1 -F
```

Point

After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

Failback of the Primary Server

Revert the primary server and standby server to the original server configuration. Do this to execute the main job on the previous primary server. Refer to "4.1.1.3 Failback of the Primary Server" for details.

Note

Obtain a backup as soon as this task is complete.

3.8.2 Stopping for Maintenance

Perform this procedure to stop all servers for periodic inspections, for example. On the server on which Mirroring Controller is running, execute the mc_ctl command in stop mode to stop the instance and Mirroring Controller.

After that, on the server where the Mirroring Controller arbitration process is running, execute the mc_arb command in stop mode to stop the Mirroring Controller arbitration process.

Stopping Mirroring Controller

Example)

```
> mc_ctl stop -M D:\mcdir\inst1 -a
```

Stopping the Mirroring Controller arbitration process

Linux

Example)

```
$ mc_arb stop -M /mcarb_dir/arbiter1
```

Windows

Example)

```
> mc_arb stop -M D:\mcarb_dir\arbiter1
```

3.8.3 Arbitration Server Maintenance

Arbitration server maintenance can be performed without affecting database server operation.

Follow the procedure below to perform arbitration server maintenance.

1. Execute the mc_arb command in stop mode to forcibly stop the Mirroring Controller arbitration process.

   Linux

   Example)

   ```
   $ mc_arb stop -M /mcarb_dir/arbiter1 -e
   ```

   Windows

   Example)

   ```
   > mc_arb stop -M D:\mcarb_dir\arbiter1 -e
   ```

2. Perform maintenance tasks.

3. Execute the mc_arb command in start mode to restart the Mirroring Controller arbitration process.

   Linux

   Example)

   ```
   $ mc_arb start -M /mcarb_dir/arbiter1
   ```

   Windows

   Example)

   ```
   > mc_arb start -M D:\mcarb_dir\arbiter1
   ```

4. Execute the mc_arb command in status mode to check that the arbitration server is connected to the database server.

   The example below executes the mc_arb command, and shows the status.

   Linux

   Example)

   ```
   $ mc_arb status -M /mcarb_dir/arbiter1

   server_id    host            status
   ---------------------------------------
   server1      192.0.3.100     online
   server2      192.0.3.110     online
   ```

   Windows

   Example)

   ```
   > mc_arb status -M D:\mcarb_dir\arbiter1

   server_id    host            status
   ---------------------------------------
   ```

```
server1      192.0.3.100     online
server2      192.0.3.110     online
```

5. Check the command output.

   Items to be checked

   Check that the output status is "online" on both lines.

3.9 Changes in Operation

The following changes in operation may be required:

- Changes required when the standby server is stopped
- Changing from single server mode to database multiplexing mode
- Changing from database multiplexing mode to single server mode
- Changing to database multiplexing mode when the arbitration server is used for automatic degradation
- Changing parameters
- Uninstalling in the database multiplexing mode

3.9.1 Changes Required when the Standby Server is Stopped

Operation when the standby server is stopped

Before performing maintenance for the primary server instance when the standby server has been stopped, stop Mirroring Controller on the primary server, comment out the synchronous_standby_names parameter in the postgresql.conf file of the primary server, and then execute the pg_ctl command in reload mode.

If this operation is not performed, operations performed on the primary server for the instance will remain in a wait state.

See

Refer to "pg_ctl" in Reference for information on the command.

Standby server downtime

If you specified the synchronous_standby_names parameter of the postgresql.conf file and then the standby server instance is stopped, consider the points below.

- The wal_sender_timeout parameter in the postgresql.conf file

  If the standby server is stopped after the timeout time set in this parameter was exceeded, an error stating that the transaction log could not be received may be output to the primary server event log, and all transaction logs that should be transferred to the standby server are accumulated.

- The wal_keep_size parameter in the postgresql.conf file

  If a transaction log that exceeds the value set in this parameter was generated while the standby server was stopped, the transaction log may be deleted.

  Additionally, setting this parameter requires consideration regarding stabilization of the database multiplexing mode. Refer to "2.11.1 Tuning to Stabilize the Database Multiplexing Mode" for details.

Note

The standby server must be rebuilt if the pending transaction log to be transferred to the standby server is lost when the standby server is started after the maintenance task is complete.

Take the action advised in the recovery operation that starts from "4.1.1.1.3 Identify cause of error and perform recovery" through to "4.1.1.2 Rebuild the Standby Server".

3.9.2 Changing from Single Server Mode to Database Multiplexing Mode

The procedure for switching single server mode to database multiplexing mode for the purposes of high reliability and load distribution of the system is explained below.

This procedure is equivalent to the setup procedure explained in "Chapter 2 Setting Up Database Multiplexing Mode".

Note

If the data storage destination directory name is not comprised of ASCII characters

Stop the application job and then perform the following procedure to migrate to a directory with a name that uses only ASCII characters:

1. Stop the database instance on the primary server.
2. Change the name of the data storage destination directory to one that uses only ASCII characters.
   For example, do not perform operations that will cause the symbolic link contained in the data storage destination directory to become lost, such as moving data to another drive.

See

When encrypting the storage data, refer to "Database Multiplexing Mode" in the Operation Guide, and then perform the setup for encryption on the primary and standby servers.

1. Install on the arbitration server

   Perform this step only if the arbitration server is used for automatic degradation.

   Install the Server Assistant on the server where the Mirroring Controller arbitration process is started.

   Refer to "Installation" in the Installation and Setup Guide for Server Assistant for information on how to install the Server Assistant.

2. Install on the standby server

   Install Fujitsu Enterprise Postgres on the server to be started as the standby server.

   Refer to "Installation" in the Installation and Setup Guide for Server for information on how to install Fujitsu Enterprise Postgres.

   Use ASCII characters in the data storage destination directory.

3. Stop the application jobs

   Stop the application jobs to be connected to the primary server.

4. Change the primary server settings

   To allow connections from the server to be started as the standby server, configure the settings in step 2 and thereafter of "2.4.2 Creating, Setting, and Registering the Primary Server Instance" on the primary server.

5. Set up the arbitration server

   Refer to "2.3 Setting Up the Arbitration Server" for details.

   Perform this step only if the arbitration server is used for automatic degradation.

6. Set up database multiplexing mode on the primary server

   Refer to "2.4.1 Setting Up Database Multiplexing Mode on the Primary Server" for details.

7. Set up database multiplexing mode on the standby server

   Refer to "2.5.1 Setting Up Database Multiplexing Mode on the Standby Server" for details.

8. Create the standby server instance and start it

   Refer to "2.5.2 Creating, Setting, and Registering the Standby Server Instance" for details.

After the above steps are completed, refer to the remaining explanations in "Chapter 2 Setting Up Database Multiplexing Mode" and ensure that the required settings and operations are completed.

3.9.3 Changing from Database Multiplexing Mode to Single Server Mode

The procedure for stopping database multiplexing mode and changing to single server mode is explained below.

Some tasks must be performed on the database server, and others must be performed on the arbitration server.

The tasks on the arbitration server are required only if the arbitration server is used for automatic degradation.

Tasks on the database server

1. Determine the server for which the instance is to be stopped, and switch this server

   Determine the server that is to be excluded as the database multiplexing mode target, and for which the instance is to be stopped.

   If the server for which the instance is to be stopped is the primary server, execute the mc_ctl command in the switch mode to switch the standby server to the primary server.

   The standby server after the switch is complete will be the server for which the instance is to be stopped.

   If the server for which the instance is to be stopped is the standby server, there is no need to perform the switch operation.

   Example)

```
> mc_ctl switch -M D:\mcdir\inst1
```

2. Stop Mirroring Controller and the instance.

   On the server that was determined in step 1, execute the mc_ctl command in the stop mode to stop Mirroring Controller and the instance.

   Example)

```
> mc_ctl stop -M D:\mcdir\inst1
```

3. Unregister Mirroring Controller from the Windows service.

   Execute the mc_ctl command in unregister mode to unregister Mirroring Controller from the Windows service.

   Example)

```
> mc_ctl unregister -M D:\mcdir\inst1
```

4. Delete registries related to the event log

   If error logs are output to the event log in "2.2.1.2 Preparatory Tasks for the Output of Error Logs to the Event Log", delete the registered event source name for each instance.

   Example)

```
> regsvr32 /u /i:"Mirroring Controller inst1" "c:\Program Files\Fujitsu\fsepv<x>server64\lib
\mcevent.dll"
```

   Note that "<*x*>" indicates the product version.

5. Delete the file resources

   Delete the following file resources:

   - Data storage destination directory

- Mirroring Controller management directory

  Example)

```
> rmdir /S /Q D:\database\inst1
> rmdir /S /Q D:\mcdir\inst1
```

See

Refer "Security-Related Notes" in the Operation Guide for information on deleting the data securely.

6. Stop the application jobs

   Stop the application jobs to be connected to the primary server.

7. Stop Mirroring Controller and the instance on the primary server

   Execute the mc_ctl command in stop mode on the primary server.

   Example)

```
> mc_ctl stop -M D:\mcdir\inst1
```

8. Unregister Mirroring Controller from the Windows service on the primary server

   Execute the mc_ctl command in unregister mode to unregister Mirroring Controller from the Windows service.

   Example)

```
> mc_ctl unregister -M D:\mcdir\inst1
```

9. Delete registries related to the event log on the primary server

   If error logs are output to the event log in "2.2.1.2 Preparatory Tasks for the Output of Error Logs to the Event Log", delete the registered event source name for each instance.

   Example)

```
> regsvr32 /u /i:"Mirroring Controller inst1" "c:\Program Files\Fujitsu\fsepv<x>server64\lib
\mcevent.dll"
```

   Note that "<x>" indicates the product version.

10. Delete the database multiplexing mode settings that were configured for the primary server instance.

    Reset the postgresql.conf file parameters to their values before the database multiplexing operation was set.

    Delete the file resources from the Mirroring Controller management directory.

    If the backup operation was performed, delete the following resources:

    - Mirroring Controller management directory backup data obtained in database multiplexing mode
    - Instance backup data obtained in database multiplexing mode

    Additionally, if the primary_conninfo parameter is set in the postgresql.auto.conf file, execute the ALTER SYSTEM RESET statement to delete the setting.

    Example)
    An example execution of the psql command is shown below.

```
postgres=# ALTER SYSTEM RESET primary_conninfo;
```

    After these actions are performed, ensure that the backup data is collected when starting the single operation.

See

- Refer to "Security-Related Notes" in the Operation Guide for details on deleting the data securely.
- Refer to "2.14 Backup Operation" for details on the backup operation.
- Refer to "Appendix A Parameters" for details on the postgresql.conf file parameters.

Point

In the above procedure, if the postgresql.conf file of the single primary server can be changed by reloading the file, the operation mode can be changed without stopping the application job.

In that case, execute the mc_ctl command in stop mode with the --mc-only option specified to stop only Mirroring Controller in relation to stopping the primary server.

Tasks on the arbitration server

Linux

1. Execute the mc_arb command in stop mode to stop the Mirroring Controller arbitration process.

   Example)

   ```
   $ mc_arb stop -M /mcarb_dir/arbiter1
   ```

2. Delete the Mirroring Controller arbitration process management directory.

   Example)

   ```
   $ rm -rf /mcarb_dir/arbiter1
   ```

Windows

1. Execute the mc_arb command in stop mode to stop the Mirroring Controller arbitration process.

   Example)

   ```
   > mc_arb stop -M D:\mcarb_dir\arbiter1
   ```

2. Unregister the Mirroring Controller arbitration process from the Windows service.

   Execute the mc_arb command in unregister mode to unregister the Mirroring Controller arbitration process from the Windows service.

   Example)

   ```
   > mc_arb unregister -M D:\mcarb_dir\arbiter1
   ```

3. Delete registrations related to the event log

   If error logs are output to the event log in "2.2.2.1 Preparing to Output Error Logs to the Event Log (Windows)", delete the registered event source name for each instance.

   Example)

   ```
   > regsvr32 /u /i:"Mirroring Controller arbtier1" "c:\Program Files\Fujitsu
   \fsepv<x>assistant64\lib\mcarbevent.dll"
   ```

   Note that "<x>" indicates the product version.

4. Delete the Mirroring Controller arbitration process management directory.

   Example)

```
> rmdir /S /Q D:\mcarb_dir\arbiter1
```

3.9.4 Changing to Database Multiplexing Mode when the Arbitration Server is Used for Automatic Degradation

This section provides the procedure to change to database multiplexing mode using the Mirroring Controller only on the database server when the arbitration server is used for automatic degradation.

Some tasks must be performed on the database server, and others must be performed on the arbitration server.

Tasks on the arbitration server

1. Set up the arbitration server.

   Refer to "2.3 Setting Up the Arbitration Server" for information on how to set up the arbitration server.

Tasks on the database server

1. On the server where Mirroring Controller is running, execute the mc_ctl command in stop mode to stop Mirroring Controller on the primary server and standby server.

   Example)

```
> mc_ctl stop -M D:\mcdir\inst1 -a --mc-only
```

2. Edit the network.conf file of the primary server and standby server to add the information of the arbitration server.

   Refer to "A.3 Network Configuration File" for details.

   The definition example of the network.conf file of the primary server is shown below:

   Example)

   The IDs of the primary server and standby server are set to "server1" and "server2", and their port numbers are set to "27540" and "27541". The ID of the server of the Mirroring Controller arbitration process is set to "arbiter", and its port number is set to "27541".

```
server1 192.0.2.100,192.0.3.100 27540,27541 server
server2 192.0.2.110,192.0.3.110 27540,27541 server
arbiter 192.0.3.120 27541 arbiter
```

Note

- Ensure that the port numbers set for the primary server, standby server, and arbitration server do not conflict with other software. Also do not configure the same segment for the admin network and arbitration network.
- If the server type is "server", two IP addresses or host names, and two port numbers need to be specified in the following order:
  - IP address or host name of the database server used as the admin network
  - IP address or host name of the database server used as the arbitration network
  - Port number of the database server used as the admin network
  - Port number of the database server used as the arbitration network
- If the server type is "arbiter", specify the IP address or host name set for the my_address parameter and the port number set for the port parameter in arbitration.conf.

3. Edit the *serverIdentifier*.conf file of the primary server and standby server to add parameters required for the operation where the arbitration server is used for automatic degradation.

   Refer to "A.4.1 Server Configuration File for the Database Servers" for information on the parameters required when the arbitration server is used for automatic degradation.

4. On the primary server and standby server, execute the mc_ctl command in start mode to start the Mirroring Controller process.

   Example)

```
> mc_ctl start -M D:\mcdir\inst1 --mc-only
```

Common tasks

1. Check the connection status from the database server or arbitration server.

   Refer to "2.8 Checking the Connection Status" for details.

3.9.5 Changing Parameters

Stop Mirroring Controller before editing the Mirroring Controller server configuration file and network configuration file.

If the Mirroring Controller process crashes or becomes unresponsive, restart is performed automatically by the Mirroring Controller monitoring process, and the configuration file is reloaded. Therefore, if the configuration file was being edited, unintended behavior will occur.

3.9.6 Uninstalling in Database Multiplexing Mode

This section explains how to uninstall Fujitsu Enterprise Postgres on a server using database multiplexing mode.

Some tasks must be performed on the database server, and others must be performed on the arbitration server.

The tasks on the arbitration server are required only if the arbitration server is used for automatic degradation.

Tasks on the database server

1. Stop the multiplexed instances and Mirroring Controller

   Refer to "3.2 Starting and Stopping Mirroring Controller" for information on how to stop the instance.

2. Unregister Mirroring Controller from the Windows service

   Execute the mc_ctl command in unregister mode to unregister Mirroring Controller from the Windows service.

   Example)

```
> mc_ctl unregister -M D:\mcdir\inst1
```

3. Delete registries related to the event log

   If messages are output to the event log, DLLs are registered in accordance with "2.2.1.2 Preparatory Tasks for the Output of Error Logs to the Event Log". Delete these registries so that no unnecessary issues occur.

   Example)
   The following is an example in which the DLL of a 64-bit product that is registered under the default event source name is deleted. Note that "<*x*>" indicates the product version.

```
> regsvr32 /u "c:\Program Files\Fujitsu\fsepv<x>server64\lib\mcevent.dll"
```

   - Delete the registered DLL for each instance

     DLL registration is performed so that messages output to the event log are identified by each instance, and are output to any event source named by the user.

     Accordingly, it is necessary to delete the DLL registry for each instance. Delete the DLL registry for each event source name.

     Example)
     The following is an example in which the DLL of a 64-bit product that is registered under the

event source name "Mirroring Controller inst1" is deleted. Note that "<*x*>" indicates the product version.

```
> regsvr32 /u /i:"Mirroring Controller inst1" "c:\Program Files\Fujitsu\fsepv<x>server64\lib
\mcevent.dll"
```

- If installing multiple versions

  If the database multiplexing system you set up using the Fujitsu Enterprise Postgres package has been set to output error logs to the event log, use the DLL path name that you took note of previously as explained in "2.2.1.2 Preparatory Tasks for the Output of Error Logs to the Event Log" to reregister the default event source name.

Note

Ensure that you delete the DLLs before the uninstallation. If you perform the uninstallation without doing so, you may not be able to delete the DLLs at a later time.

4. Uninstall Fujitsu Enterprise Postgres

   Refer to "Uninstallation" in the Installation and Setup Guide for Server for information on how to uninstall Fujitsu Enterprise Postgres.

Tasks on the arbitration server

Refer to "Uninstallation" in the Installation and Setup Guide for Server Assistant, and uninstall the Server Assistant.

Chapter 4 Action Required when an Error Occurs in Database Multiplexing Mode

This chapter describes the action required if an error occurs in database multiplexing mode.

In database multiplexing mode, when an error is detected, the switch or disconnection of the standby server is performed automatically, so that only the primary server starts degrading. In this case, the recovery tasks will be required for the standby server on which the switch or disconnection was performed.

Other possible cases are as follows:

- When automatic switch fails
- When automatic disconnection fails
- When all servers or instances were stopped

4.1 Action Required when Server Degradation Occurs

If the server has started degrading, the recovery tasks will vary depending on whether the cause was the switch (failover or switchover), or the disconnection.

Execute the mc_ctl command in status mode, or refer to the event log, and check if the cause of the server to start degrading was the switch or the disconnection.

In the example below, the mc_ctl command is executed in status mode.

If a switch has occurred, "switched" (the switch is complete and the server is in a degrading state) is displayed for "mirroring status".

Example)

```
> mc_ctl status -M D:\mcdir\inst1
mirroring status
----------------
switched

   :
```

If a disconnection has occurred, "not-switchable" (disconnection was performed so the server cannot be switched) is displayed for "mirroring status".

Example)

```
> mc_ctl status -M D:\mcdir\inst1
mirroring status
----------------
not-switchable

  :
```

Note

If Mirroring Controller detects any errors on the server on which operations are continuing during recovery to database multiplexing mode from a degrading operation state, perform the procedure in "4.1.3 Addressing Errors During Degrading Operation", and then recover to database multiplexing mode.

4.1.1 Operations when the Server has Started Degrading after a Switch has Occurred

This section explains the operations when the server has started degrading after a switch has occurred.

Note

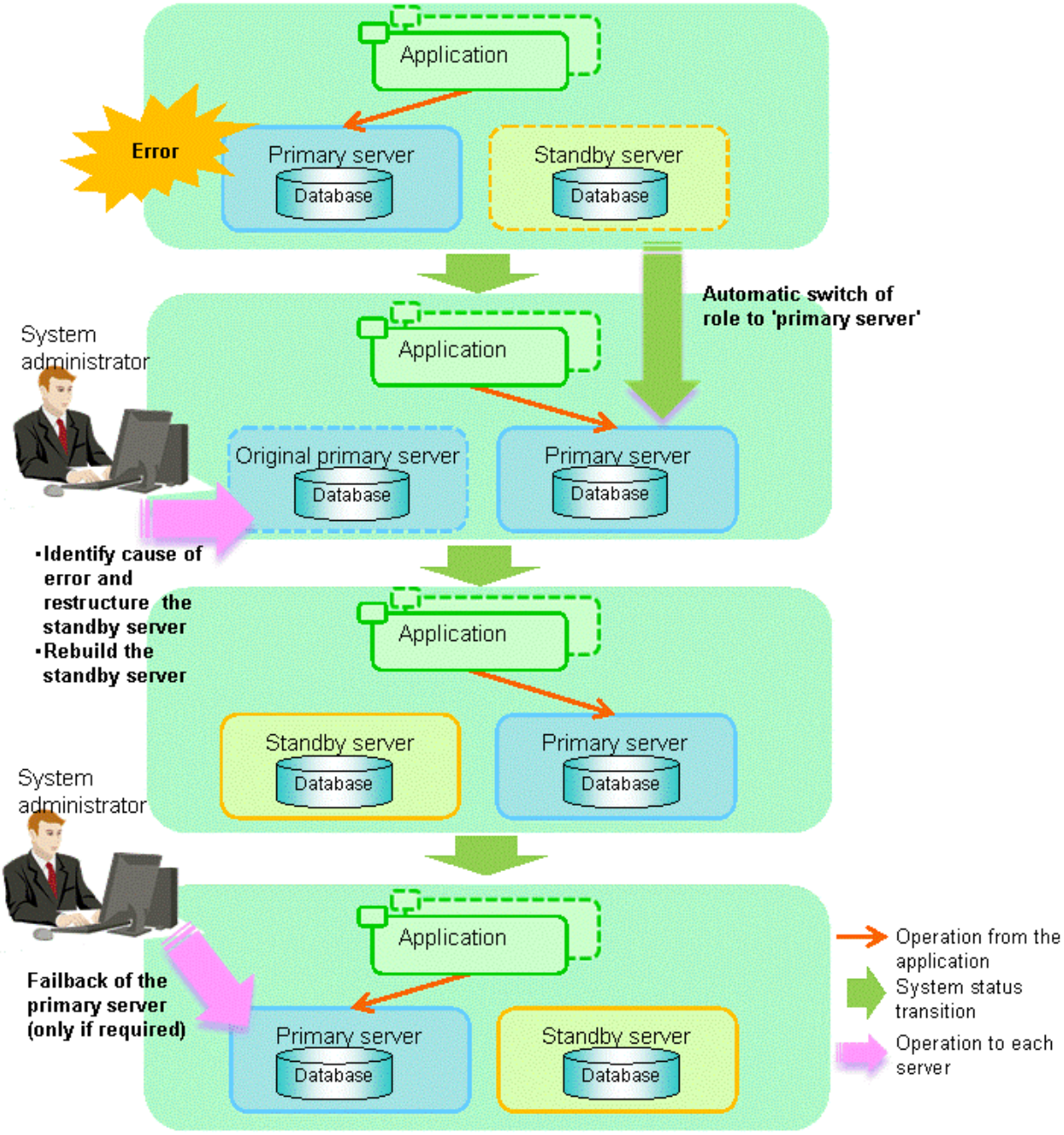
- After a switch has occurred as a result of an abnormality on the primary server, the database will not have a multiplexed configuration until the standby server is rebuilt. Remove the cause of the error as quickly as possible, and then rebuild the standby server.
- If the reference job was executed on the standby server, and the servers are switched because an error occurred on the primary server, the load is concentrated on the new primary server. Accordingly, pause the reference job on the original standby server, rebuild the original primary server as the new standby server, and then resume the reference job for the new standby server.
- If the instance on the new primary server is stopped before the original primary server where the error occurred is rebuilt as the new standby server, a split brain occurs at startup from the instance on the original primary server. Therefore, start the instance on the new primary server before rebuilding the standby server.

If the switch occurred and the server has started degrading, perform the following operations to recover the standby server and revert it to its original state:

- Identify Cause of Error and Restore the Standby Server
- Rebuild the Standby Server
- Failback of the Primary Server (only if required)

The flow of these operations is shown in the figure below.

Figure 4.1 Flow of operations



4.1.1.1 Identify Cause of Error and Restore the Standby Server

Perform the recovery according to the following procedure:

1. Stop Mirroring Controller
2. Recovery of the Mirroring Controller management directory
3. Identify cause of error and perform recovery

4.1.1.1.1 Stop Mirroring Controller

Execute the mc_ctl command in stop mode for the original primary server on which the error occurred.

```
Example)
```

```
> mc_ctl stop -M D:\mcdir\inst1
```

This also stops the instance that is required to perform the recovery.

Note

If the instance does not stop, refer to "Actions in Response to Failure to Stop an Instance" in the Operation Guide, and then stop the instance.

Then, specify the -e option in the above command to forcibly stop Mirroring Controller.

4.1.1.1.2 Recovery of the Mirroring Controller management directory

Copy the files in the Mirroring Controller management directory from the backup data, and then perform the recovery.

4.1.1.1.3 Identify cause of error and perform recovery

Refer to the event log of the primary server and the standby server to identify the cause of the error, and then perform recovery.

The following commands can be used to recover a standby server. Select depending on the recovery and the situation.

- pg_basebackup

  Creates a copy of all resources of the primary server instance.

- pg_rewind

  Creates a copy of only the updated files on the new primary server. For this reason, if this command is used to incorporate a new standby server, recovery time can be shortened. To use this command to build the original primary server as a new standby server, at least one of the following must be met:

  a. Checksums were enabled when an instance was created, or

  b. The wal_log_hints parameter of postgresql.conf was enabled when an instance was started.

  Additionally, full_page_writes must be enabled, which is its default value.

See

- Refer to "pg_basebackup" in "Reference" in the PostgreSQL Documentation for information on the pg_basebackup command.
- Refer to "pg_rewind" in "Reference" in the PostgreSQL Documentation for information on the pg_rewind command.

The example below executes the pg_rewind command to perform recovery by synchronizing data on the original primary server with the new primary server.

1. Wait for the application of unapplied update transaction logs on the new primary server.

   Execute the SQL below on the new primary server, and wait until the result is false.

   ```
   # select pg_is_in_recovery();
   ```

   Example)

   ```
   > psql -h hostNameOfNewPrimaryServer -p portNumOfNewPrimaryServer -d dbName -c "select
   pg_is_in_recovery();"
   ```

   Any database can be connected to.

Note

If the pg_rewind command is executed immediately after promotion of the new primary server, the processing in steps 1 and 2 is required. If update-type SQL can be executed on the new primary server and checkpoint processing is executed after promotion, the processing in steps 1 and 2 will not be necessary.

2. Update the timeline ID.

   Execute checkpoint processing, and update the timeline ID.

   ```
   > psql -h hostNameOfNewPrimaryServer -p portNumOfNewPrimaryServer -d dbName -c "checkpoint;"
   ```

   Any database can be connected to.

3. Create a copy of the new primary server instance in the original primary server (new standby server).

   Execute the pg_rewind command to synchronize the new standby server data with the new primary server.

   Example)

   ```
   > pg_rewind -D D:\database\inst1 -R --source-server="user=userName host=newPrimaryServerHostName
   port=newPrimaryServerPortNumber dbname=dbName application_name=newStandbyServerName"
   ```

   ### Note

   - Use the pg_rewind command with the -R option to create a standby.signal file. If you do not create the standby.signal file, the Mirroring Controller cannot be started as a standby server.
   - If using a method that requires password authentication for connections to the primary server, you will need to ensure that authentication is performed automatically. If the -R option is specified for the pg_rewind command and the password parameter is specified for the --dbname option, the pg_rewind command will set the password in the primary_conninfo parameter in postgresql.auto.conf file, enabling connections to be performed automatically.

     If a password is not set in the primary_conninfo parameter in postgresql.auto.conf file, it will be necessary to create a password file (%APPDATA%postgresql\pgpass.conf), and then specify a password for the replication database.
   - If you need to set a connection string other than host, port and application_name, include it in the setting of the primary_conninfo parameter.
   - The primary_conninfo parameter should not be set in the postgresql.conf file, but only in the postgresql.auto.conf file using the pg_rewind command.

4. Specify parameters in the postgresql.conf file of the original primary server (new standby server).

   Set the parameters required for the standby server in postgresql.conf.

   Refer to "Table 2.5 Parameters" for information on the parameters to set in postgresql.conf.

See

- Refer to "Hot Standby" in the PostgreSQL Documentation for details on the standby.signal file.
- Refer to "Setting Up a Standby Server" in the PostgreSQL Documentation for details on the primary_conninfo.

Note

A new timeline is branched for the new primary server due to promotion, so 'latest' needs to be specified for the recovery_target_timeline parameter so that the old primary server (new standby server) follows the new primary server.

4.1.1.2 Rebuild the Standby Server

The starting of the recovered original primary server as the standby server is referred to as the "standby server rebuild".

On the original primary server, start Mirroring Controller and the instance.

Enabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode.

Example)

```
> mc_ctl start -M D:\mcdir\inst1
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

Example)

```
> mc_ctl start -M D:\mcdir\inst1 -F
```

Point

After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

4.1.1.3 Failback of the Primary Server

To revert the primary server and standby server to the original server configuration after rebuilding the standby server, perform failback for the primary server.

Do this to execute the main job on the previous primary server.

Perform the following procedure:

1. Failback of the primary server

   Execute the mc_ctl command in switch mode on the primary server or the standby server.

   Example)

   ```
   > mc_ctl switch -M D:\mcdir\inst1
   ```

   After executing the mc_ctl command in switch mode, the status will be as follows:

   Example)

   ```
   > mc_ctl status -M D:\mcdir\inst1
   mirroring status
   ----------------
   switched
   server_id  host_role                  host        host_status   db_proc_status       disk_status
   ------------------------------------------------------------------------------------------------
   server1    primary                  192.0.2.100 normal        abnormal(postmaster) normal
   server2    none(inactivated primary) 192.0.2.110 normal        abnormal(postmaster) normal
   ```

2. Stop the original primary server

   On the original primary server, execute the mc_ctl command in stop mode to stop Mirroring Controller and the instance.

   Example)

   ```
   > mc_ctl stop -M D:\mcdir\inst1
   ```

3. Create a copy of the new primary server instance in the original primary server (new standby server)

   Execute the pg_basebackup command to create data in the new standby server by synchronizing with the new primary server.

   Example)

```
> pg_basebackup -D D:\database\inst1 -X fetch --waldir=E:\transaction\inst1 --progress --verbose
-R --dbname="application_name=standbyServerName" -h primaryServerHostName -p
primaryServerPortNumber
```

   See

   The procedure for copying the new primary server instance to the new standby server is the same as the procedure for setting up the new standby server.

   Refer to "2.5.2 Creating, Setting, and Registering the Standby Server Instance", and then perform the recovery.

4. Rebuild the standby server

   On the standby server, start Mirroring Controller and the instance.

   Enabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode.

   Example)

```
> mc_ctl start -M D:\mcdir\inst1
```

   Disabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

   Example)

```
> mc_ctl start -M D:\mcdir\inst1 -F
```

   Point

   After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

4.1.2 Operations when the Server has Started Degrading after a Disconnection has Occurred

This section explains the operations when the server has started degrading after a disconnection has occurred.

Note

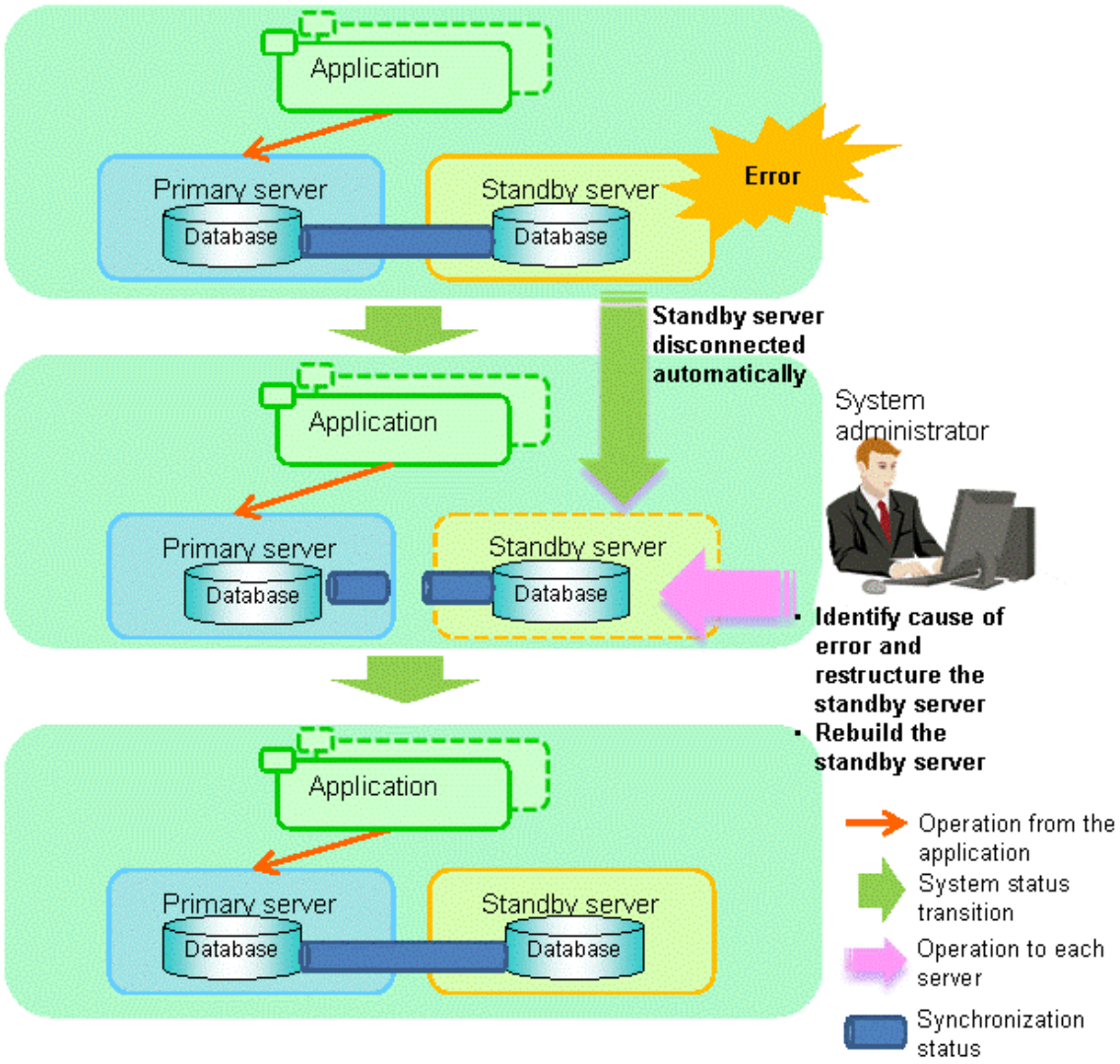
After a disconnection has occurred as a result of an abnormality on the standby server, the database will not have a multiplexed configuration until the standby server is rebuilt. Remove the cause of the error as quickly as possible, and then rebuild the standby server.

If the disconnection occurred and the server has started degrading, perform the following operations to recover the standby server and revert it to its original state:

- Identify Cause of Error and Restore the Standby Server
- Rebuild the Standby Server

The flow of these operations is shown in the figure below.

Figure 4.2 Flow of operations



4.1.2.1 Identify Cause of Error and Restore the Standby Server

Perform the recovery according to the following procedure:

1. Stop Mirroring Controller
2. Recovery of the Mirroring Controller management directory
3. Identify cause of error and perform recovery

4.1.2.1.1 Stop Mirroring Controller

Execute the mc_ctl command in stop mode for the standby server on which the error occurred.

Example)

```
> mc_ctl stop -M D:\mcdir\inst1
```

This also stops the instance that is required to perform the recovery.

Note

If the instance does not stop, refer to "Actions in Response to Failure to Stop an Instance" in the Operation Guide, and then stop the instance.

Then, specify the -e option in the above command to forcibly stop Mirroring Controller.

4.1.2.1.2 Recovery of the Mirroring Controller management directory

Copy the files in the Mirroring Controller management directory from the backup data, and then perform the recovery.

4.1.2.1.3 Identify cause of error and perform recovery

Refer to the event log of the primary server and the standby server to identify the cause of the error, and then perform recovery.

Execute the pg_basebackup command to perform recovery by synchronizing data in the primary server with the standby server.

```
Example)
```

```
> pg_basebackup -D D:\database\inst1 -X fetch --waldir=E:\transaction\inst1 --progress --verbose -R --
dbname="application_name=standbyServerName" -h primaryServerHostName -p primaryServerPortNumber
```

See

This recovery procedure is the same as the procedure for setting up the standby server.

Refer to "2.5.2 Creating, Setting, and Registering the Standby Server Instance", and then perform the recovery.

4.1.2.2 Rebuild the Standby Server

Start the Mirroring Controller and the instance of the standby server, and rebuild the standby server.

Enabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode.

```
Example)
```

```
> mc_ctl start -M D:\mcdir\inst1
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

```
Example)
```

```
> mc_ctl start -M D:\mcdir\inst1 -F
```

Point

After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

4.1.3 Addressing Errors During Degrading Operation

This section explains how to address errors that may occur on the server on which operation is continuing during degrading operation triggered by a switch or disconnection.

If needing to recover from backup data

If it is necessary to recover the database using backup data due to data becoming corrupted from disk failure or user operation error, refer to the following for information on recovery to database multiplexing mode:

- Action Required when All Database Servers or Instances Stopped
- Recovering from an Incorrect User Operation

If a temporary error occurs

If a temporary error occurs, such as due to a high load on the server or insufficient system resources, remove the cause of the error and restart Mirroring Controller, and then refer to the following for details on recovery to database multiplexing mode:

- Operations when the Server has Started Degrading after a Switch has Occurred
- Operations when the Server has Started Degrading after a Disconnection has Occurred

See

Refer to "3.2.1 Starting Mirroring Controller" and "3.2.2 Stopping Mirroring Controller" for information on restarting Mirroring Controller.

4.2 Action Required when Automatic Switch Fails

If the system behavior is unstable, for example there are insufficient temporary system resources, the Mirroring Controller automatic switch may fail.

Perform the switch manually using one of the following methods:

- Refer to the procedures in "3.4 Manually Switching the Primary Server".
- In the standby server, execute the mc_ctl command in switch mode with the -force option specified to forcibly perform the switch.

  Example)

```
> mc_ctl switch -M D:\mcdir\inst1 --force
```

Point

- Even if connection cannot be established between database servers, it is possible to fence the primary server and forcibly switch by executing the mc_ctl command in switch mode with the --force option specified.
- The primary server is not fenced in the cases below, so stop Mirroring Controller and instances of the primary server database in advance:
  - The --no-fencing option is specified when performing forced switch.
  - The heartbeat_error_action parameter in *serverIdentifier*.conf is set to "message" and the fencing command is not configured to be used (the fencing_command parameter is omitted in *serverIdentifier*.conf).
  - The heartbeat_error_action parameter in *serverIdentifier*.conf is set to "fallback".

See

Recovery to database multiplexing mode

Refer to "4.1.1.2 Rebuild the Standby Server" and "4.1.1.3 Failback of the Primary Server" for information on recovery to database multiplexing mode.

4.3 Action Required when Automatic Disconnection Fails

If the system behavior is unstable, for example there are insufficient system resources such as available memory or free disk space, automatic disconnection using Mirroring Controller may not be possible.

Perform the disconnection manually using one of the following methods:

- Refer to the procedures in "3.5 Manually Disconnecting the Standby Server".
- In the primary server, execute the mc_ctl command in detach mode to perform forced disconnection.

  Example)

```
> mc_ctl detach -M D:\mcdir\inst1
```

Point

- Even if connection cannot be established between database servers, it is possible to fence the standby server and forcibly disconnect by executing the mc_ctl command in detach mode.
- In the cases below, stop Mirroring Controller and instances of the standby server database in advance so that the standby server is not fenced:
  - The --no-fencing option is specified when performing forced disconnection.
  - The heartbeat_error_action parameter in *serverIdentifier*.conf is set to "message" and the fencing command is not configured to be used (the fencing_command parameter is omitted in *serverIdentifier.conf*).
  - The heartbeat_error_action parameter in *serverIdentifier*.conf is set to "fallback".

See

Recovery to database multiplexing mode

Refer to "4.1.2.2 Rebuild the Standby Server" for information on recovery to database multiplexing mode.

4.4 Action Required when All Database Servers or Instances Stopped

This section explains what happens when all database servers or instances on the database server have stopped, so jobs cannot continue.

See

Recovery to database multiplexing mode

Refer to "4.1.1.2 Rebuild the Standby Server" and "4.1.1.3 Failback of the Primary Server" for information on recovery to database multiplexing mode.

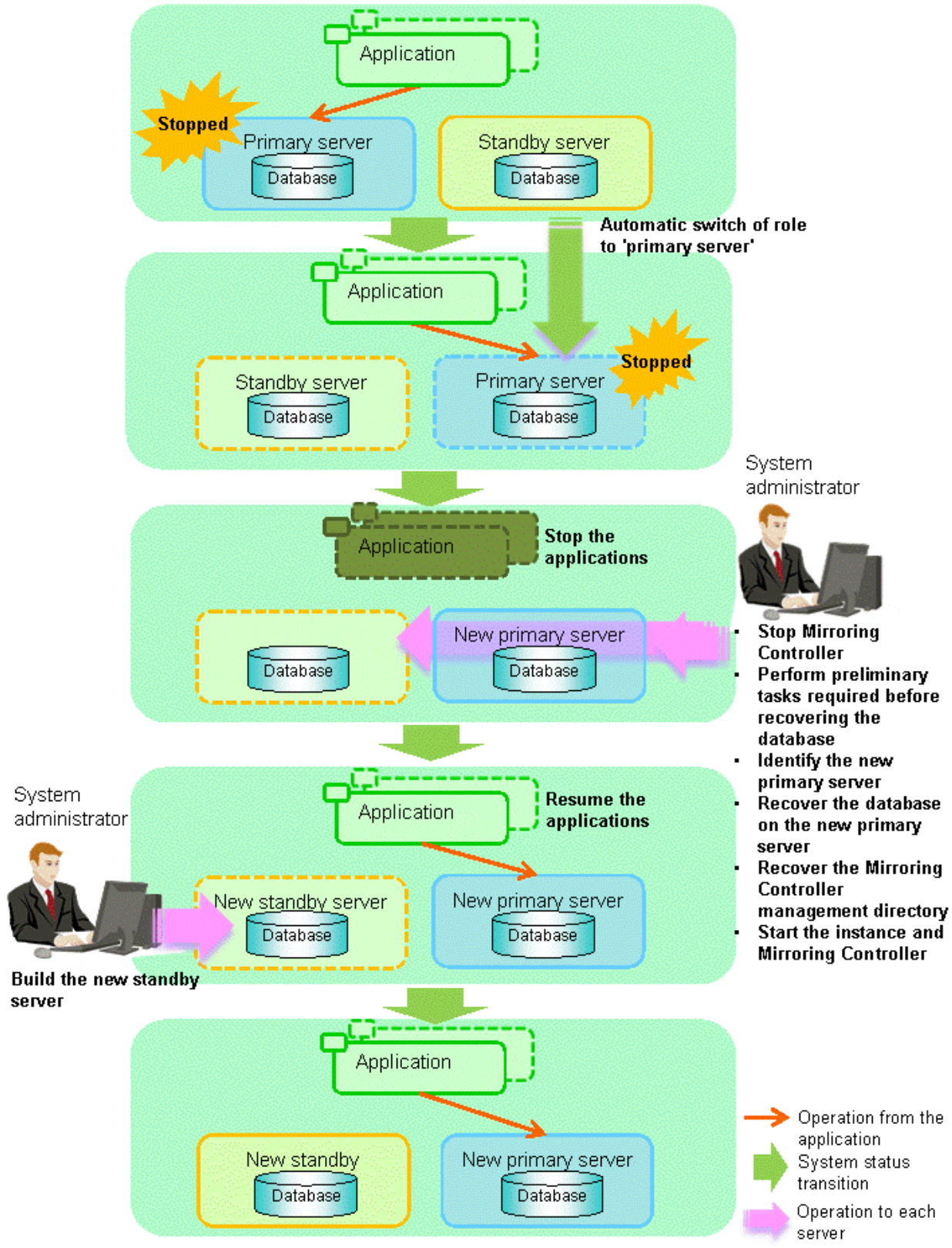
Overview of recovery operations

After recovering the database to the state immediately prior to the failure on a specific server comprising the database multiplexing system, restore the system.

In other words, after specifying the server on which the database is to be recovered and then recovering it as the new primary server, configure all other servers as new standby servers.

The flow of these recovery operations is shown in the figure below.

Figure 4.3 Flow of recovery operations



Perform the following procedure.

1. Stop applications

   Stop running applications.

2. Stop Mirroring Controller

   Execute the mc_ctl command in stop mode on all servers that comprise the database multiplexing system.

   Example)

```
> mc_ctl stop -M D:\mcdir\inst1
```

   Note

   Forcibly stop Mirroring Controller

   If Mirroring Controller does not stop, execute the mc_ctl command in stop mode with the -e option specified.

   Example)

```
> mc_ctl stop -M D:\mcdir\inst1 -e
```

3. Perform prerequisite tasks before recovering the database

   First, refer to "Actions when an Error Occurs" in the Operation Guide, and then identify the cause of the error and perform recovery of the disk on which the failure occurred, etc.

4. Identify the new primary server

   Perform the following operations on all servers comprising the database multiplexing system, and check the server containing the backup data that shows the latest date. This server will become the new primary server, on which the database is to be recovered.

   Example)
   In the example below, the pgx_rcvall command is executed with the -l option specified and the backup data that shows the latest date is identified.

```
> pgx_rcvall -l -D D:\database\inst1
Date                    Status          Dir
2013-07-01 13:30:40     COMPLETE        E:/backup/inst1/2013-07-01_13-30-40
```

5. Recover the database on the new primary server

   Recover the database using the recovery method that uses the pgx_rcvall command based on the backup data.

   a. Perform the following operations on all servers comprising the database multiplexing system, and check the server containing the archive log and mirrored transaction log that show the latest date.

      Example)
      In the example below, the archive log and mirrored transaction log that show the latest date are identified.

```
> dir /OD <backupDataStorageDir>\*_wal
```

   b. If the server containing the latest archive log and mirrored transaction log is different to the new primary server identified in step 4, all files and directories under the directory shown below are copied and written to the backup storage destination directory on the new primary server.

      Deployment destination directory of the archive log and mirrored transaction log

```
<backupDataStorageDir>\*_wal
```

   c. Execute the pgx_rcvall command on the new primary server, specifying the backup storage destination directory of the new primary server.

      Example)
      In the example below, the pgx_rcvall command is executed with the -B option specified.

```
> pgx_rcvall -B E:\backup\inst1 -D D:\database\inst1
```

See

Refer to "Actions when an Error Occurs" in the Operation Guide for information on the pgx_rcvall command.

6. Recover the Mirroring Controller management directory

   Copy the files in the Mirroring Controller management directory from the backup data on the new primary server, and then perform the recovery.

7. Start the instance and Mirroring Controller

   Start the instance and Mirroring Controller on the new primary server.

   Enabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode.

   Example)

```
> mc_ctl start -M D:\mcdir\inst1
```

   Disabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

   Example)

```
> mc_ctl start -M D:\mcdir\inst1 -F
```

   Point

   After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

8. Resume applications

   Resume execution of applications.

9. Build the new standby server

   Refer to "2.5 Setting Up the Standby Server" for information on building (setting up) a standby server from the primary server.

Point

It is not necessary to repeat steps that have already been performed, such as registering to Windows services.

4.5 Recovering from an Incorrect User Operation

This section describes how to recover an instance when data has been corrupted due to incorrect user operation.

For example, when data has been corrupted due to incorrect user operation, such as data being unintentionally changed or deleted by an application or command, it is necessary to restore the original data on the primary server and resynchronize with the standby server.

Use the following procedure to perform recovery.

1. Identify the primary server

   Execute the mc_ctl command in status mode on each server, and search for a server for which "primary" or "none(inactivated primary)" is displayed.

2. Stop the applications and commands that caused the incorrect operation to occur

   Stop applications and commands that are running on the primary server. This will minimize the impact caused by the incorrect data.

Also, if any applications used for reference by the standby server are running, stop them too.

3. Stop the instance and Mirroring Controller

   Stop the instance and Mirroring Controller on both the primary server and standby server.

   Example)

```
$ mc_ctl stop -a -M D:\mcdir\inst1
```

4. Recover the database on the primary server

   Recover the database using the recovery method in which the pgx_rcvall command uses the backup data to recover the database to a restore point prior to the time when the incorrect operation was performed.

See

Refer to "Recovering from an Incorrect User Operation" in the Operation Guide for information on using the pgx_rcvall command to recover the database to a restore point, and then perform only the database recovery procedure while the instance is in a stop state.

5. Start the instance and Mirroring Controller

   Start the instance and Mirroring Controller on the primary server.

   Enabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode.

   Example)

```
> mc_ctl start -M D:\mcdir\inst1
```

   Disabling automatic switch/disconnection

   As the instance administrator user, execute the mc_ctl command in start mode with the -F option specified.

   Example)

```
> mc_ctl start -M D:\mcdir\inst1 -F
```

Point

After Mirroring Controller is started, automatic switch/disconnection can be enabled or disabled using the enable-failover or disable-failover mode of the mc_ctl command.

6. Build the new standby server

   Refer to "2.5 Setting Up the Standby Server" for information on building (setting up) a standby server from the primary server.

Point

It is not necessary to repeat steps that have already been performed, such as registering to Windows services.

Chapter 5 Managing Mirroring Controller Using WebAdmin

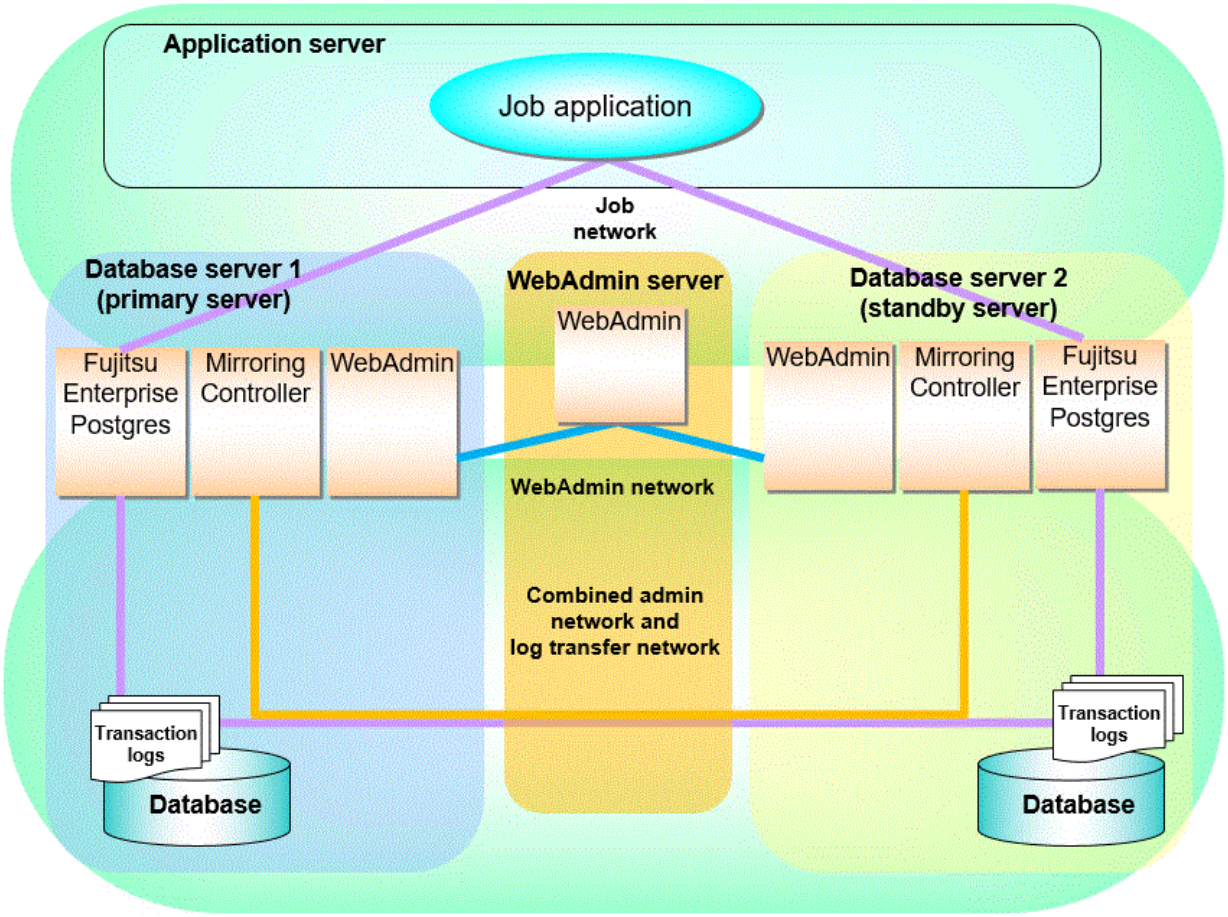
This chapter describes how to set up and manage Mirroring Controller in a streaming replication cluster using WebAdmin.

Mirroring Controller can be used to monitor a streaming replication cluster and perform automatic switching or disconnect synchronous replication when there is an error.

WebAdmin can be used to set up Mirroring Controller in an existing replication cluster. Mirroring Controller can be set up for either synchronous standby instances or asynchronous standby instances.

The configuration of the database multiplexing system built using WebAdmin is shown below:

Figure 5.1 Configuration of database multiplexing operation system using WebAdmin



Point

- If Mirroring Controller is set up to the replication cluster using WebAdmin, the network with the host name (or IP address) specified in [Host name] will be used as the admin network and the log transfer network.
- To use a network other than the job network as the log transfer network, before building the replication cluster specify a host name other than the job network one in [Host name].

Note

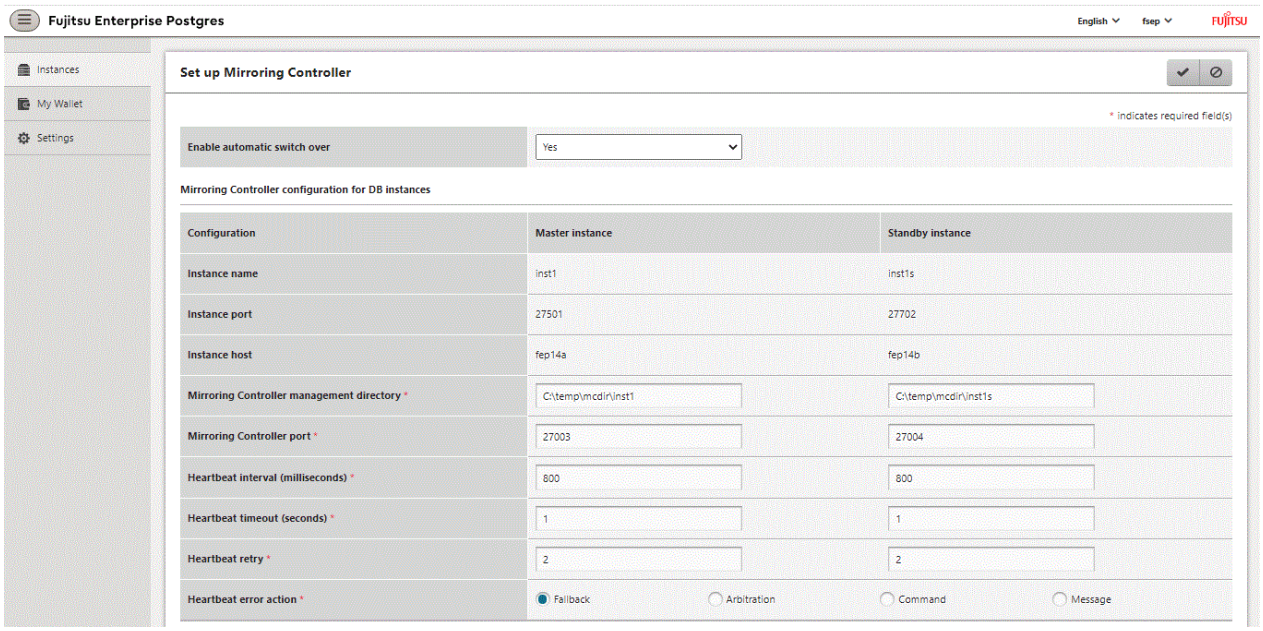
If you set up the arbitration server using WebAdmin, install WebAdmin on the arbitration server.

5.1 Mirroring Controller Setup

Perform the following procedure to set up Mirroring Controller in a streaming replication cluster.

1. In the [Instances] tab, select the standby instance on which Mirroring Controller needs to be set up.
2. Click .
3. Enter the information for the Mirroring Controller to be set up.

   In the example below, Mirroring Controller is being set up for the replication cluster having master instance "inst1" and standby instance "inst1s".



   The instance name, host address and port of the master and standby instances are displayed for easy reference.

   Enter the following items on master instance and on standby instance fields for Mirroring Controller setup, as shown in the above screenshot:

   - [Enable automatic switch over]: Toggles the automatic switch/disconnection functionality. Select "Yes". The default is "No".
   - [Mirroring Controller management directory]: Directory where the Mirroring Controller configuration files will be stored. When the [Mirroring Controller management directory] is entered, WebAdmin will search the Mirroring Controller configuration files in the entered directory based on the [Data storage path] of the corresponding DB instance. If Mirroring Controller configuration files are found, the Mirroring Controller fields will be auto filled.
   - [Mirroring Controller port]: Port number of Mirroring Controller. Note that if the Windows firewall feature is enabled, you must enable the port number of Mirroring Controller. Refer to "E.2 Windows Firewall Settings" for details.
   - [Heartbeat interval (milliseconds)]: Number of milliseconds between two consecutive heartbeat checks. The default is "800".
   - [Heartbeat timeout (seconds)]: Number of seconds for the heartbeat timeout. The default is "1".
   - [Heartbeat retry]: Number of retries for heartbeat monitoring, before failover occurs. The default is "2".
   - [Heartbeat error action]: Operation when a heartbeat abnormality is detected. The default is "Fallback".

     When using FUJITSU Enterprise Postgres 10 or 11 instance which was created with FUJITSU Enterprise Postgres WebAdmin 10 or 11, the mode in the [Heartbeat error action] for Mirroring Controller setup cannot be changed.

     When setting up Mirroring Controller for FUJITSU Enterprise Postgres 9.5 and 9.6 instances, the [Heartbeat error action] is not supported and therefore is not displayed.

   When the [Heartbeat error action] is set to "Arbitration", the following extra items are displayed:

- [Arbitration network IP address]: IP address of the arbitration network.
- [Mirroring Controller Arbitration port]: Port number of Mirroring Controller for communicating with the arbitration server.

The [Arbitration server configuration] section is also displayed with the following items. The [Arbitration server configuration] will not be auto filled.

- [Location]: Location of the arbitration server. "Local" or "Remote" can be selected depending on your configuration.

  If the arbitration server and WebAdmin server are located on the same server, you can select "Local" and the following items are displayed:

  - [Arbitration management directory]: Directory where the arbitration server configuration files will be stored.
  - [Arbitration server host or IP address]: Host name or IP address of the arbitration server.
  - [Arbitration process port]: Port number for the arbitration process.
  - [Fencing command]: Full path of the fencing command that fences a database server when an abnormality is detected.

  If "Remote" is set for the item, the items below are displayed in addition to the above items.

  - In the [Arbitration server configuration] section, [Operating system credential] is displayed where you can enter the following information:

    [User name]: User name to access the arbitration server.

    [Password]: Password to access the arbitration server.

  - In the [Remote WebAdmin for Arbitration server] section, the following items are displayed:

    [Remote WebAdmin address]: IP address of the remote WebAdmin installed on the arbitration server.

    [Remote WebAdmin port]: Port number for the WebAdmin installed on the arbitration server.

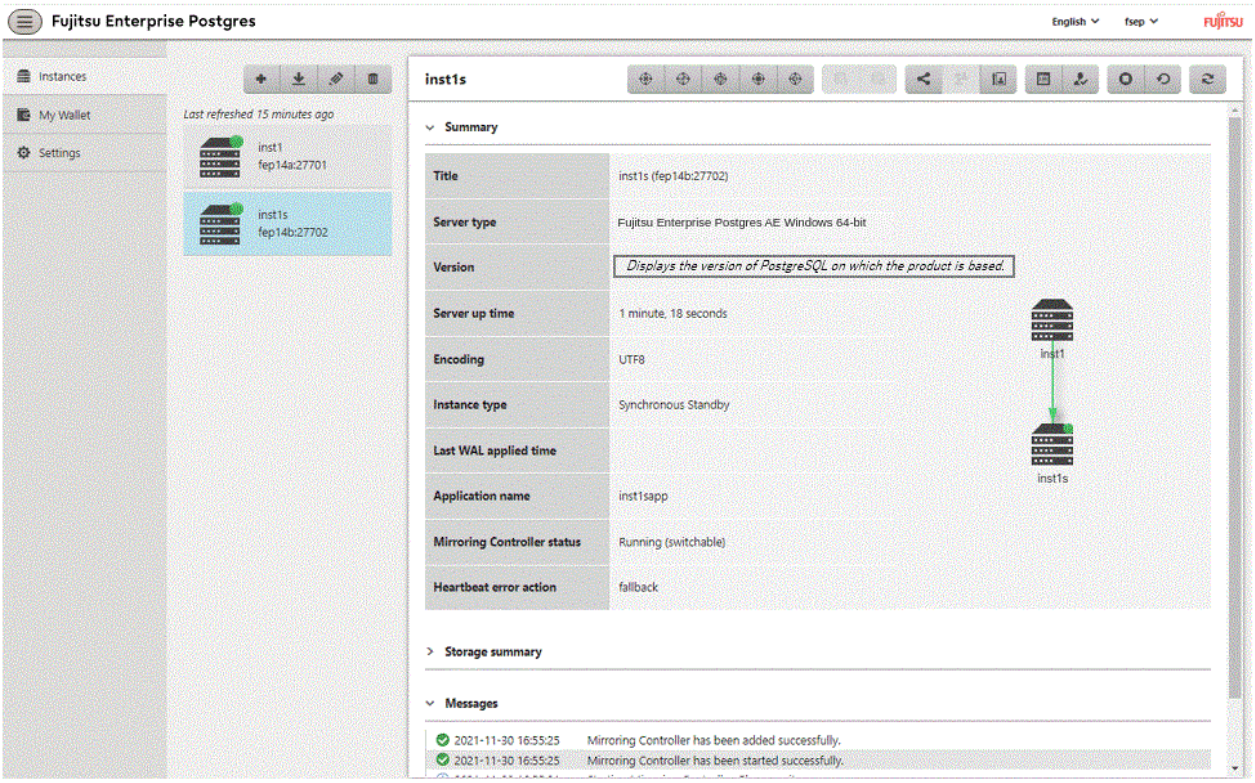
When the [Heartbeat error action] is set to "Command", the following extra items are displayed:

- [Arbitration command]: Full path of the arbitration command to be executed when an abnormality is detected.
- [Fencing command]: Full path of the fencing command that fences a database server when an abnormality is detected.

4. Click to set up Mirroring Controller.

5. Upon successful completion, Mirroring Controller will be started on master and standby instances.

   In the [Instances] tab, select standby instance. The page below is displayed, in which you can check the Mirroring Controller status. In the example below, standby instance "inst1s" is used.



After the Mirroring Controller has been set up, ([Edit Mirroring Controller] button) and ([Mirroring Controller Configuration] button) are available. These buttons are displayed only when FUJITSU Enterprise Postgres 10 or later instances are created with FUJITSU Enterprise Postgres WebAdmin 12 or later.

For FUJITSU Enterprise Postgres 9.5 and 9.6 instances, the [Heartbeat error action] will not be displayed.

When the [Heartbeat error action] is "Arbitration", the following information is displayed: whether the arbitration status is "online" or "offline", the arbitration server IP address and the arbitration process port.

Note

Operating system credential (User name, Password) should not contain hazardous characters. Refer to "Appendix F WebAdmin Disallow User Inputs Containing Hazardous Characters".

5.2 Edit Mirroring Controller Setup

Settings made in "5.1 Mirroring Controller Setup" can be updated in either the master instance or a standby instance using WebAdmin.

Perform the following procedure to edit Mirroring Controller configuration:

1. In the [Instances] tab, select the instance for which the Mirroring Controller configuration is to be edited.
2. Click .
3. Enter the information for the Mirroring Controller to be updated. Refer to "5.1 Mirroring Controller Setup".
4. Click to update the Mirroring Controller.
5. Upon successful completion, Mirroring Controller will be started on master and standby instances.

Editing and saving the [Edit Mirroring Controller] page will reset all other settings that are not listed on this page to default values.

5.3 Mirroring Controller Configuration

The information related to Mirroring Controller monitoring and control (refer to "A.4.1 Server Configuration File for the Database Servers") and the information related to arbitration and control of the Mirroring Controller arbitration process (refer to "A.4.2 Arbitration Configuration File") can be set using WebAdmin. You can view and update the configuration on either the master instance or the standby instance.

Perform the following procedure:

1. In the [Instances] tab, select the instance for the Mirroring Controller configuration you want to view.
2. Click to view the Mirroring Controller configuration.
3. Click to show the editing page for the Mirroring Controller configuration. The Mirroring Controller configurations defined during [Mirroring Controller Setup] are read-only on this page. Refer to "5.1 Mirroring Controller Setup".

   Additionally, refer to the "Appendix A Parameters" for information about the settings and the corresponding parameter names.

   The items common to all [Heartbeat error action] are:

   - Target DB
   - Core file path
   - Remote call timeout (milliseconds)
   - Agent alive timeout (seconds)
   - DB instance check interval (milliseconds)
   - DB instance check timeout (seconds)
   - DB instance check retry
   - DB instance timeout action
   - Disk check interval (milliseconds)
   - Disk check retry
   - Tablespace directory error action
   - Post-switch command
   - Post-promote command
     (Post-promote command is replaced in FUJITSU Enterprise Postgres 12. The Post-promote command is still valid and will be displayed when it is used in the server configuration file of Mirroring Controller.)
   - Post-attach command
   - Pre-detach command
   - State transition command timeout (seconds)
   - Check synchronous standby names validation

   When the [Heartbeat error action] is set to "Arbitration", the following extra items are displayed:

   - Arbitration timeout (seconds)
   - Arbiter alive interval (milliseconds)
   - Arbiter alive retry
   - Arbiter alive timeout (seconds)
   - Arbiter connect interval (milliseconds)

- Arbiter connect timeout (seconds)
- Fencing command
- Fencing command timeout (seconds)
- Shutdown detached synchronous standby

When the [Heartbeat error action] is set to "Arbitration", the [Arbitration server configuration] section is displayed with the following items:

- Core file path
- Fencing command timeout (seconds)
- Heartbeat interval (milliseconds)
- Heartbeat timeout (seconds)
- Heartbeat retry

When the [Heartbeat error action] is set to "Command", the following extra items are available:

- Fencing command timeout (seconds)
- Arbitration command timeout (seconds)
- Shutdown detached synchronous standby

When the [Heartbeat error action] is set to "Message", the following extra items are available:

- Fencing command
- Fencing command timeout (seconds)

In addition, the following configurations are provided:

- DB instance JDBC connection SSL parameters
- DB instance libpq connection SSL parameters

4. Click to update the Mirroring Controller configurations.

5.4 Stopping Mirroring Controller

Mirroring Controller can be stopped either in master instance or in standby instance using WebAdmin.

Perform the following procedure to stop Mirroring Controller.

1. In the [Instances] tab, select the instance where to stop Mirroring Controller.
2. Click .
3. In the confirmation dialog box, click [Yes].

Mirroring Controller will be stopped on the selected instance. The Mirroring Controller status will be updated, and a confirmation message entry will be displayed in the [Message] section.

5.5 Starting Mirroring Controller

Mirroring Controller can be started either in master instance or in standby instance using WebAdmin.

Perform the following procedure to start Mirroring Controller.

1. In the [Instances] tab, select the instance where to start Mirroring Controller.
2. Click .
3. In the confirmation dialog box, select the desired failover mode, and then click [Yes].

Mirroring Controller will be started on the selected instance. The Mirroring Controller status will be updated, and a confirmation message entry will be displayed in the [Message] section.

5.6 Disabling Failover Mode

Disabling failover mode in Mirroring Controller disables automatic switch/disconnection between master and standby instances.

Perform the following procedure to disable failover mode.

1. In the [Instances] tab, select the instance.
2. Click .
3. In the confirmation dialog box, click [Yes].

Failover mode will be disabled in Mirroring Controller. The Mirroring Controller status will be updated and a confirmation message entry will be displayed in the [Message] section.

5.7 Enabling Failover Mode

Enabling failover mode in Mirroring Controller enables automatic switch/disconnection between master and standby instances.

Perform the following procedure to enable failover.

1. In the [Instances] tab, select the instance.
2. Click .
3. In the confirmation dialog box, click [Yes].

Failover mode will be enabled in Mirroring Controller. The Mirroring Controller status will be updated and a confirmation message entry will be displayed in the [Message] section.

5.8 Deleting Mirroring Controller Setup

Deleting Mirroring Controller setup removes its setup from master and standby instances.

1. In the [Instances] tab, select the instance.
2. Click .
3. In the confirmation dialog box, click [Yes].

Mirroring Controller setup will be removed from the cluster. The cluster status will be updated and a confirmation message entry will be displayed in the [Message] section.

For the instances in FUJITSU Enterprise Postgres 12 or later, WebAdmin does not delete the Mirroring Controller management directory and the configuration files.

5.9 Status Update after Failover

When Mirroring Controller performs a failover, standby instance will be promoted to standalone instance. The Mirroring Controller setup will be removed from both standby and master instances.

The following scenario describes one of the ways in which failover can be triggered, and the results achieved by the use of Mirroring Controller in WebAdmin.

1. In the [Instances] tab, select the master instance "inst1".

2. Click .
3. In the confirmation dialog box, the warning "This instance is being monitored by Mirroring Controller. Stopping the instance may result in cluster failover." is displayed.
4. Choose the stop mode and click [Yes].

   In the server, the following takes place:

   a. The master instance is stopped.
   b. Failover is triggered in Mirroring Controller.
   c. The Mirroring Controller setup is removed from both master and standby instances
   d. Standby instance is promoted to standalone.
5. When the instance is refreshed in WebAdmin, the latest status of the instances will be displayed.

Note

When failover is performed, the Mirroring Controller setup is removed from both master and standby instances. Therefore, to manage the Mirroring Controller using WebAdmin again, create the standby instance and set up Mirroring Controller.

Refer to "Creating a Standby Instance" in the Operation Guide for details.

Refer to "5.1 Mirroring Controller Setup" for details.

5.10 Action Required when an Error Occurs in the Combined Admin Network and Log Transfer Network

Communication errors may temporarily occur in the network used as the admin network and log transfer network due to reasons such as high load on the server or insufficient system resources. Because of this, there is a risk of causing a split-brain situation by mistake even though the server has no issues.

Split brain is a phenomenon in which both servers temporarily operate as primary servers, causing data updates to be performed on both servers.

How to detect split brain using WebAdmin

If the conditions below are met, split brain may occur. Refer to "Split-brain detection method" and "How to recover from a split-brain" in "Appendix D Notes on Performing Automatic Degradation Immediately after a Heartbeat Abnormality" and take the actions described.

1. A standby instance is selected in the [Instances] tab, and
2. "Standalone" is displayed in [Instance type], and
3. A master instance is selected in the [Instances] tab, and
4. "Standalone" is displayed in [Instance type].

Note

The admin network is important because Mirroring Controllers use it to confirm the status of each server.

The log transfer network is also important to maintain the data freshness.

Therefore, use network configurations resistant to faults for these networks by using the network redundancy channel bonding feature provided by the operating system or network driver vendor.

5.11 Performing Automatic Degradation Using the Arbitration Server

If database multiplexing is performed using WebAdmin, it is also possible to perform automatic degradation using the arbitration server. In such cases, it is necessary to perform tasks on the database server and the arbitration server after setting up Mirroring Controller in WebAdmin.

Tasks on the arbitration server

Perform setup of the arbitration server using Mirroring Controller commands.

1. Set up the arbitration server.

   Refer to "2.3 Setting Up the Arbitration Server" in "Chapter 2 Setting Up Database Multiplexing Mode" for information on how to set up the arbitration server.

Tasks on the database server

Change some of the settings after setting up Mirroring Controller in WebAdmin.

1. Set up Mirroring Controller in WebAdmin.

   Refer to "5.1 Mirroring Controller Setup" for details.

2. Use WebAdmin to stop Mirroring Controller on the master and standby instances.

   Refer to "5.4 Stopping Mirroring Controller" for details.

3. Edit the network configuration file of the master and standby instances, and add the arbitration server information.

   The network configuration file is network.conf, which exists in the Mirroring Controller management directory specified during Mirroring Controller setup. Refer to "A.3 Network Configuration File" for details.

   A definition example of network.conf is shown below.

   ```
   Example:
   The port number of the database server to be used as the arbitration network is set to "27541".
   The ID of the server of the Mirroring Controller arbitration process is set to "arbiter", and its
   port number is set to "27541".
   ```

   ```
   dbsvm27500 192.0.2.100,192.0.3.100 27540,27541 server
   dbsvs27500 192.0.2.110,192.0.3.110 27540,27541 server
   arbiter 192.0.3.120 27541 arbiter
   ```

Note

- Ensure that the port numbers set for the database server and the arbitration server do not conflict with other software. In addition, do not configure the same segment for the admin network and the arbitration network.
- If the server type is "server", two IP addresses or host names, and two port numbers need to be specified in the following order:
  - IP address or host name of the database server used as the admin network
  - IP address or host name of the database server used as the arbitration network
  - Port number of the database server used as the admin network
  - Port number of the database server used as the arbitration network
- If the server type is "arbiter", specify the IP address or host name set for the my_address parameter and the port number set for the port parameter in arbitration.conf of the arbitration server.
- WebAdmin also support editing mirroring controller configuration via Use WebAdmin to edit Mirroring Controller configurations.
  Refer to "5.2 Edit Mirroring Controller Setup" for detailes.

4. Edit the server configuration file of the master and standby instances, and add the parameters required for automatic degradation using the arbitration server.

   The server configuration file is *instanceName*.conf or *instancePort*.conf, which exists in the Mirroring Controller management directory specified during Mirroring Controller setup.

   To perform automatic degradation using the arbitration server, set the heartbeat_error_action parameter to "arbitration".

   Refer to "A.4.1 Server Configuration File for the Database Servers" for information on other parameters.

5. Use WebAdmin to start Mirroring Controller on the master and standby instances.

   Refer to "5.5 Starting Mirroring Controller" for details.

Common tasks

1. Use the Mirroring Controller command to check the connection status from the database server or the arbitration server.

   Refer to "2.8 Checking the Connection Status" for information on how to check the connection status.

Appendix A Parameters

This appendix describes the configuration files and parameters required by the database multiplexing mode.

See

Refer to "Server Configuration" in the PostgreSQL Documentation for information on the postgresql.conf file.

A.1 Parameters Set on the Primary Server

The content for the parameters set in the postgresql.conf file of the primary server is shown in the table below.

Table A.1 postgresql.conf file

Parameter	Value set	Explanation
wal_level	replica or logical	Specify the output level for the transaction log. Specify "logical" when logical decoding is also to be used.
max_wal_senders	2 or more	Specify "2" when building a Mirroring Controller cluster system. When additionally connecting asynchronous standby servers to the cluster system, add the number of simultaneous connections from these standby servers.
synchronous_standby_names	'*standbyServerName*'	Use single quotation marks (') to enclose the name that will identify the standby server. Any name can be specified. Do not change this parameter while Mirroring Controller is running. Do not specify multiple names to this parameter as the Mirroring Controller can manage only one standby server.
hot_standby	on	Specify whether queries can be run on the standby server. Specify this to execute reference jobs on the standby server.
wal_keep_size	WAL save size (megabytes)	If a delay exceeding the value set in this parameter occurs, the WAL segment required later by the primary server may be deleted. Additionally, if you stop a standby server (for maintenance, for example), consider the stop time and set a value that will not cause the WAL segment to be deleted. Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on estimating the WAL save size.
wal_log_hints	on	When using the pg_rewind command to recover a standby server, specify this parameter or enable checksums when executing the initdb command.
wal_sender_timeout	Timeout (milliseconds)	Specify the time period after which it is determined that the receiver process (walreceiver) of the transaction log is in an abnormal state on the primary server. The specified value must be larger than the value set for the wal_receiver_status_interval parameter set in the postgresql.conf file of the standby server.

Parameter	Value set	Explanation
		By aligning this value with the value for the database process heartbeat monitoring time, you can unify the time after which it is determined that an error has occurred.
wal_receiver_timeout	Timeout (milliseconds)	Specify the time period after which it is determined that an error has occurred when the transaction log was received on the standby server. By aligning this value with the value for the database process heartbeat monitoring time, you can unify the time after which it is determined that an error has occurred.
archive_mode	on	Specify the archive log mode.
archive_command	'cmd /c ""installDir\\bin\ \pgx_walcopy.cmd" "%p" "backupDataStorageDestinationDir \\archived_wal\\%f"""	Specify the command and storage destination to save the transaction log.
backup_destination	Backup data storage destination directory	Specify the name of directory where to store the backup data. Set the permissions so that only the instance administrator user can access the specified directory. Specify the same full path on all servers, so that the backup data of other servers can be used to perform recovery.
listen_addresses	Primary server IP address, host name, or "*"	Specify the IP address or host name of the primary server. Specify the IP address or corresponding host name that will be used to connect to the log transfer network. The content specified is also used to allow connections from client applications. To receive the connection and the transaction log from any client or standby server, specify "*". Refer to "Connections and Authentication" in the PostgreSQL Documentation for details.
max_connections	Number of simultaneous client connections to the instance + superuser_reserved_connections value	The value specified is also used to restrict the number of connections from client applications and the number of connections for the management of instances. Refer to "When an Instance was Created with the initdb Command" in the Installation and Setup Guide for Server, and "Connections and Authentication" in the PostgreSQL Documentation, for details.
superuser_reserved_connections	Add the number of simultaneous executions of mc_ctl status (*1) + 2	Specify the number of connections reserved for connections from database superusers. Add the number of connections from Mirroring Controller processes. Also reflect the added value in the max_connections parameter.
restart_after_crash	off	If "on" is specified, or the default value is used for this parameter, behavior equivalent to restarting Fujitsu Enterprise Postgres, including crash recovery, will be performed when some server processes end abnormally. However, when database multiplexing monitoring is used, a failover will occur after an error is detected when some server processes end abnormally, and the restart of those server processes is forcibly stopped. Specify "off" to

Parameter	Value set	Explanation
		prevent behavior such as this from occurring for no apparent reason.
synchronous_commit	on or remote_apply	Specify up to what position WAL send is to be performed before transaction commit processing returns a normal termination response to a client. The recommended value is "on" or "remote_apply" to prevent data loss caused by operating system or server down immediately after a switch or switch.
recovery_target_timeline	latest	Specify "latest" so that the new standby server (original primary server) will follow the new primary server when a switch occurs. This parameter is required when the original primary server is incorporated as a new standby server after the primary server is switched.

*1: Number of simultaneous executions of the mc_ctl command in the status mode.

A.2 Parameters Set on the Standby Server

This section explains the content of the file and parameters set on the standby server. After editing postgresql.conf file, start the instance.

The content for the parameters specified in postgresql.conf file is shown in the table below.

Table A.2 postgresql.conf file

Parameter	Value set	Explanation
wal_level	replica or logical	Specify the output level for the transaction log. Specify "logical" when logical decoding is also to be used.
max_wal_senders	2 or more	Specify "2" when building a Mirroring Controller cluster system. When additionally connecting asynchronous standby servers to the cluster system, add the number of simultaneous connections from these standby servers.
synchronous_standby_names	'*primaryServerName*'	Use single quotation marks (') to enclose the name that will identify the primary server. Any name can be specified. This name will be required to rebuild the original primary server as the new standby server after the primary server was switched. Do not change this parameter while Mirroring Controller is running. Do not specify multiple names to this parameter as the Mirroring Controller can manage only one standby server.
hot_standby	on	Specify whether queries can be run on the standby server. Specify this to execute reference type jobs on the standby server.

Parameter	Value set	Explanation
wal_keep_size	WAL save size (megabytes)	If a delay exceeding the value set in this parameter occurs, the WAL segment required later by the primary server may be deleted. Additionally, if you stop a standby server (for maintenance, for example), consider the stop time and set a value that will not cause the WAL segment to be deleted. Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on estimating the WAL save size.
wal_log_hints	on	When using the pg_rewind command to recover a standby server, specify this parameter or enable checksums when executing the initdb command.
wal_sender_timeout	Timeout (milliseconds)	Specify the time period after which it is determined that the receiver process (walreceiver) of the transaction log is in an abnormal state on the primary server. The specified value must be larger than the value set for the wal_receiver_status_interval parameter set in the postgresql.conf file of the standby server. By aligning this value with the value for the database process heartbeat monitoring time, you can unify the time after which it is determined that an error has occurred.
wal_receiver_timeout	Timeout (milliseconds)	Specify the time period after which it is determined that an error has occurred when the transaction log was received on the standby server. By aligning this value with the value for the database process heartbeat monitoring time, you can unify the time after which it is determined that an error has occurred.
backup_destination	Backup data storage destination directory	Specify the name of the backup data storage directory. Set the permissions so that only the instance administrator user can access the specified directory. Specify the same full path on all servers so that the backup data of other servers can be used to perform recovery.
archive_mode	on	Specify the archive log mode.
archive_command	'cmd /c ""installDir\\bin\\pgx_walcopy.cmd" "%p" "backupDataStorageDestinationDir\\archived_wal\\%f""'	Specify the command and storage destination to save the transaction log.
listen_addresses	Standby server IP address, host name, or "*"	Specify the IP address or host name of the standby server. Specify the IP address or corresponding host name that will be used to connect to the log transfer network. The content specified is also used to allow connections from client applications. To receive the connection and the transaction log from any client or standby server, specify "*".

Parameter	Value set	Explanation
		Refer to "Connections and Authentication" in the PostgreSQL Documentation for details.
max_connections	Number of simultaneous client connections to the instance + superuser_reserved_connections value	The value specified is also used to restrict the number of connections from client applications and the number of connections for the management of instances. Refer to "When an Instance was Created with the initdb Command" in the Installation and Setup Guide for Server, and "Connections and Authentication" in the PostgreSQL Documentation, for details.
superuser_reserved_connections	Add the number of simultaneous executions of mc_ctl status (*1) + 2	Specify the number of connections reserved for connections from database superusers. Add the number of connections from Mirroring Controller processes. Also reflect the added value in the max_connections parameter.
restart_after_crash	off	If "on" is specified, or the default value is used for this parameter, behavior equivalent to restarting Fujitsu Enterprise Postgres, including crash recovery, will be performed when some server processes end abnormally. However, when database multiplexing monitoring is used, a failover will occur after an error is detected when some server processes end abnormally, and the restart of those server processes is forcibly stopped. Specify "off" to prevent behavior such as this from occurring for no apparent reason.
synchronous_commit	on or remote_apply	Specify up to what position WAL send is to be performed before transaction commit processing returns a normal termination response to a client. The recommended value is "on" or "remote_apply" to prevent data loss caused by operating system or server down immediately after a switch or switch.
primary_conninfo	'*streamingReplicationConnectionDestinationInfo*'	Use single quotation marks (') to enclose the connection destination information of the streaming replication. The default value of this parameter is automatically set to postgresql.auto.conf in the procedure to run pg_basebackup for instance setup.
recovery_target_timeline	latest	Specify "latest" so that the new standby server (original primary server) will follow the new primary server when a switch occurs. This parameter is required when the original primary server is incorporated as a new standby server after the primary server is switched.

A.3 Network Configuration File

This section explains the network configuration file (network.conf) to be defined individually for the database servers and the arbitration server. Define the same content on the primary server and standby server.

For database multiplexing mode, define the network configuration for the following in network.conf.

- Integration between Mirroring Controller processes
- Integration between a Mirroring Controller process and the Mirroring Controller arbitration process

Items to be defined in network.conf

Format:

```
serverIdentifier hostName[,hostName] portNum[,portNum] [serverType]
Or,
serverIdentifier ipAddr[,ipAddr] portNum[,portNum] [serverType]
```

Specify the server identifier, IP address or host name, port number, and server type, using a space as the delimiter.

The items are explained in the table below.

Table A.3 network.conf file

Item	Description
serverIdentifier	Specify any identifier for the server. The maximum length is 64 bytes. Use ASCII characters excluding spaces and number signs (#) to specify this parameter.
ipAddrOrHostName	Specify the IP address or its corresponding host name that will connect to the admin network that performs communication between the database servers, and to the arbitration network that performs communication between a database server and the arbitration server. When specifying two IP addresses or host names delimited by a comma, do not insert a space after the comma. Use ASCII characters excluding spaces to specify the host name.
portNum	A port number cannot be specified if it exceeds the range 0 to 65535. Ensure that the port number does not conflict with other software. Do not specify an ephemeral port that may temporarily be assigned by another program. Note that the value specified in this parameter must also be set in the services file. When specifying two port numbers delimited by a comma, do not insert a space after the comma.
serverType	Specify "server" for a database server ("server" can be omitted), or "arbiter" for the arbitration server.

Content to be defined on the database servers

This section explains the network.conf content to be defined on the database servers.

The content to be defined depends on the operation settings at the time a heartbeat abnormality is detected.

When automatic degradation by the arbitration server is selected

- Specify definitions related to the admin network and arbitration network.
- Specify the IP address or host name and port number according to the server type (database server or arbitration server) as shown in the table below.

Server type	IP address or host name		Port number	
	First	Second	First	Second
server	IP address or host name used as the admin network	IP address or host name used as the arbitration network (*1)	Port number used as the admin network	Port number used as the arbitration network (*1)
arbiter	IP address or host name of the arbitration server Specify the same value as that specified in the my_address parameter of arbitration.conf on the arbitration server.	Not required	Port number on the arbitration server Specify the same value as that specified in the port parameter of arbitration.conf on the arbitration server.	Not required

*1: This value can be omitted from definitions not related to the local server. If it is omitted, network.conf must be created on both the primary server and standby server.

Example)

IPv4

```
server1 192.0.2.100,192.0.3.100 27540,27541 server
server2 192.0.2.110,192.0.3.110 27540,27541 server
arbiter 192.0.3.120 27541 arbiter
```

IPv6

```
server1 2001:258:8404:1217:250:56ff:fea7:559f,2001:258:8404:1217:250:56ff:fea8:559f
27540,27541 server
server2 2001:258:8404:1217:250:56ff:fea7:55a0,2001:258:8404:1217:250:56ff:fea8:55a0
27540,27541 server
arbiter 2001:258:8404:1217:250:56ff:fea8:55a0 27541 arbiter
```

When operation other than automatic degradation by the arbitration server is selected

- Specify definitions related to the admin network.
- Define the same content on the primary server and standby server.
- Define lines for database servers only.
- Specify only one IP address or host name and port number.

IP address or host name		Port number	
First	Second	First	Second
IP address or host name to be used as the admin network	Not required	Port number used as the admin network	Not required

Example)

The literal space represents a space.

IPv4

```
server1 192.0.2.100 27540
server2 192.0.2.110 27540
```

IPv6

```
server1 2001:258:8404:1217:250:56ff:fea7:559f 27540
server2 2001:258:8404:1217:250:56ff:fea7:55a0 27540
```

Content to be defined on the arbitration server

This section explains the network.conf content to be defined on the arbitration server.

- Specify definitions related to the arbitration network.
- Define lines for database servers only.
- For the IP address or host name, specify the same value as the second IP address or host name specified in the database server line in network.conf of the database server.

- For the port number, specify the same value as the second port number specified in the database server line in network.conf of the database server.

Example)

The literal space represents a space.

IPv4

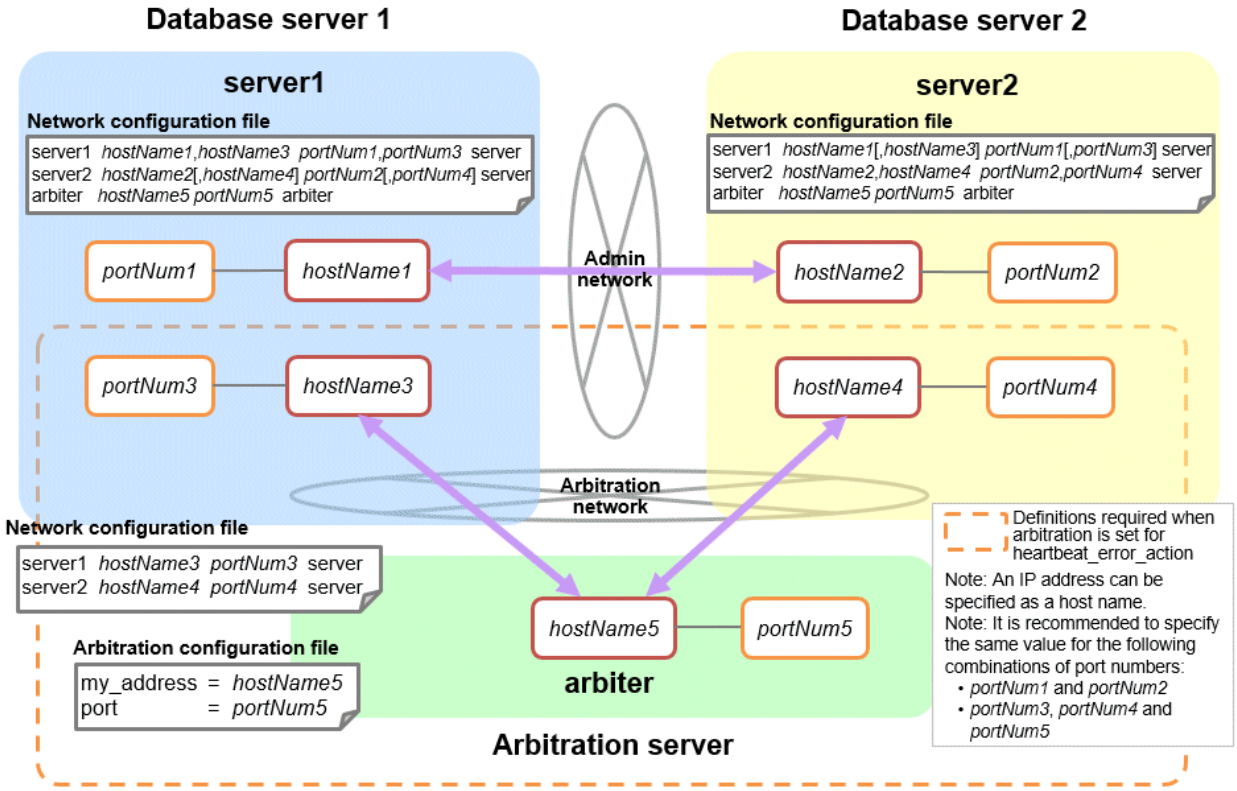
```
server1 192.0.3.100 27541
server2 192.0.3.110 27541
```

IPv6

```
server1 2001:258:8404:1217:250:56ff:fea8:559f 27541
server2 2001:258:8404:1217:250:56ff:fea8:55a0 27541
```

Relationship between network-related definitions

Refer to the diagram below for the relationship between the host names and IP addresses or port numbers specified in the network configuration file (network.conf) and arbitration configuration file (arbitration.conf).



A.4 Server Configuration File

A.4.1 Server Configuration File for the Database Servers

Define the information related to Mirroring Controller monitoring and control in the serverIdentifier.conf file. The maximum length of the server identifier is 64 bytes. Use ASCII characters excluding spaces to specify this parameter.

If the primary server and standby server environments are different, define content that is different, according to the environment.

Table A.4 serverIdentifier.conf file

Parameter	Value set	Explanation
db_instance	*'dataStorageDestinationDir'* [Example] db_instance = 'D:\\database1\\inst1'	Use halfwidth single quotation marks (') to enclose the data storage destination directory that will identify the monitoring target instance. Use ASCII characters, and specify "\\" as the path delimiter.
target_db	postgres or template1	Specify the name of the database to be connected to the database instance. The default is "postgres".
db_instance_username	'*usernameToConnectToDbInstance*'	Specify the username to connect to the database instance. Use ASCII characters to specify this parameter. Specify this parameter if the database administrator user is different from the operating system user who starts Mirroring Controller. Enclose the username of the database superuser in single quotation marks ('). The maximum length of the username is 63 bytes. The default is the operating system user who starts Mirroring Controller.
db_instance_password	'*passwordOfInstanceAdminUser*'	Specify the password used when Mirroring Controller connects to a database instance, enclosed in single quotation marks ('). Use ASCII characters to specify this parameter. If password authentication is performed, you must specify this parameter in the settings used when Mirroring Controller connects to a database instance. If you specify this parameter when password authentication is not performed, the parameter will be ignored. If the specified value of this parameter includes ' or \, write \' or \\, respectively.
enable_hash_in_password	on or off	Specify on to treat the # in the db_instance_password specification as a password character, or off to treat it as a comment. The default is "off".
core_file_path	*'coreFileOutputDir'*	Specify the directory to which the core file is to be output, enclosed in single quotation marks ('). Use ASCII characters, and specify "\\" as the path delimiter. If this parameter is omitted, it will be assumed that the Mirroring Controller management directory was specified.
db_instance_service_name	'*registeredServiceNameOfFujitsuEnterprisePostgresInstance*'	Specify the registered service name of the Fujitsu Enterprise Postgres instance in the Windows service. Use ASCII characters to specify this parameter, enclosed in single quotation marks (').
mc_service_name	'*registeredServiceNameOfMirroringController*'	Specify the Mirroring Controller service name registered in the Windows service. Use ASCII characters excluding forward slash (/) and backslash (\) to specify this parameter, enclosed in single quotation marks (').

Parameter	Value set	Explanation
		The maximum length of the service name is 124 bytes. The default is 'MirroringControllerOpen'.
event_source(*1)	'*eventSourceName*'	Specify the event source name to be used to identify the Mirroring Controller message in the event log, enclosed in single quotation marks ('). Use ASCII characters to specify this parameter. The maximum length of the event source name is 255 bytes. The default is 'MirroringControllerOpen'.
remote_call_timeout	Admin communication timeout	Specify the timeout value (milliseconds) of the Mirroring Controller agent process for communication between servers. Specify a value between 0 and 2147483647 to be less than the operation system TCP connection timeout (*2). In addition, when using the Mirroring Controller arbitrage process, fencing commands, and state transition commands, specify a value that is greater than the sum of the timeout values (*3). The value 0 indicates that there is no timeout limit. The default is 70000 milliseconds (70 seconds).
agent_alive_timeout	Timeout for Mirroring Controller process heartbeat monitoring (seconds)	If there is no response for at least the number of seconds specified, the Mirroring Controller process is restarted. Specify 0 or a value between 2 and 2147483647. The value 0 indicates that there is no timeout limit. The default is 0 seconds.
heartbeat_error_action	Operation when a heartbeat abnormality is detected using operating system or server heartbeat monitoring	arbitration: Perform automatic degradation using the arbitration server. command: Call a user exit to determine degradation, and perform automatic degradation if required. message: Notify messages. fallback: Perform automatic degradation unconditionally. The default is "arbitration". Set the same value on the primary server and standby server.
heartbeat_interval	Interval time for abnormality monitoring during heartbeat monitoring of the operating system or server (milliseconds)	Abnormality monitoring of the operating system or server is performed at the interval specified in heartbeat_interval. If an error is detected, operation will conform to the value specified for heartbeat_error_action. If "arbitration" is specified in heartbeat_error_action, the error detection time during monitoring of the operating system or server becomes longer than when the arbitration server is not used, by up to the value specified for arbitration_timeout. Specify a value between 1 and 2147483647. The specified value is used as the default for db_instance_check_interval and disk_check_interval. The default is 800 milliseconds.

Parameter	Value set	Explanation
heartbeat_timeout	Timeout for abnormality monitoring during heartbeat monitoring of the operating system or server (seconds)	If there is no response for at least the number of seconds specified, it will be assumed that an error has occurred that requires the primary server to be switched, or the standby server to be disconnected. If an error is detected, operation will conform to the value specified for heartbeat_error_action. If "arbitration" is specified in heartbeat_error_action, the error detection time during monitoring of the operating system or server becomes longer than when the arbitration server is not used, by up to the value specified for arbitration_timeout. Specify a value between 1 and 2147483647. The specified value is used as the default for db_instance_check_timeout. The default is 1 second.
heartbeat_retry	Number of retries for abnormality monitoring during heartbeat monitoring of the operating system or server (number of times)	Specify the number of retries to be performed when an error has been detected that requires the primary server to be switched, or the standby server to be disconnected. If an error is detected in succession more than the specified number of times, switch or disconnection will be performed. If an error is detected, operation will conform to the value specified for heartbeat_error_action. If "arbitration" is specified in heartbeat_error_action, the error detection time during monitoring of the operating system or server becomes longer than when the arbitration server is not used, by up to the value specified for arbitration_timeout. Specify a value between 0 and 2147483647. The specified value is used as the default for db_instance_check_retry and disk_check_retry. The default is 2 times.
db_instance_check_interval	Database process heartbeat monitoring interval (milliseconds)	Heartbeat monitoring of the database process is performed at the interval specified in db_instance_check_interval. This parameter setting is also used for abnormality monitoring of streaming replication. Specify a value between 1 and 2147483647. The default is the value set for heartbeat_interval.
db_instance_check_timeout	Database process heartbeat monitoring timeout (seconds)	If there is no response for at least the number of seconds specified, it will be assumed that an error has occurred that requires the primary server to be switched, or the standby server to be disconnected. Specify a value between 1 and 2147483647. The default is the value set for heartbeat_timeout.
db_instance_check_retry	Number of retries for database process heartbeat monitoring (number of times)	Specify the number of retries to be performed when an error has been detected that requires the primary server to be switched, or the standby server to be disconnected. If an error is detected in succession more than the specified number of times, switch or disconnection will be performed. However, if it detects that the database process

Parameter	Value set	Explanation
		is down, it will immediately switch or disconnect regardless of the setting of this parameter. This parameter setting is also used for abnormality monitoring of streaming replication. Specify a value between 0 and 2147483647. The default number of retries is the value set for heartbeat_retry.
db_instance_timeout_action	none, message, or failover	Specify the behavior for no-response monitoring of the instance. none: Do not perform no-response monitoring. message: Notify messages if an error is detected during no-response monitoring. failover: Perform automatic degradation if an error is detected during no-response monitoring. The default is "failover".
disk_check_interval	Interval time for disk abnormality monitoring (milliseconds)	Abnormality monitoring of disk failure is performed at the interval (milliseconds) specified in disk_check_interval. If the file cannot be created, it will be assumed that an error has occurred that requires the primary server to be switched, or the standby server to be disconnected. Specify a value between 1 and 2147483647. The default is the value set for heartbeat_interval.
disk_check_retry	Number of retries for disk abnormality monitoring (number of times)	Specify the number of retries to be performed when an error has been detected that requires the primary server to be switched, or the standby server to be disconnected. If an error is detected in succession more than the specified number of times, switch or disconnection will be performed. Specify a value between 0 and 2147483647. The default number of retries is the value set for heartbeat_retry.
tablespace_directory_error_action	message or failover	Specify the behavior to be implemented if an error is detected in the tablespace storage directory. message: Notify messages. failover: Perform automatic degradation. The default is "failover".
arbiter_alive_interval	Interval time for monitoring connection to the Mirroring Controller arbitration process (milliseconds)	A heartbeat is sent to the Mirroring Controller arbitration process at the specified interval. Specify a value between 1 and 2147483647. The default is 16000 milliseconds. This parameter does not need to be set for operation that does not use the arbitration server.
arbiter_alive_timeout	Timeout for monitoring connection to the Mirroring Controller arbitration process (seconds)	If the heartbeat does not respond within the specified number of seconds, the Mirroring Controller arbitration

Parameter	Value set	Explanation
		process is determined to have been disconnected, a message is output, and reconnection is attempted. Specify a value between 1 and 2147483647. The default is 20 seconds. This parameter does not need to be set for operation that does not use the arbitration server.
arbiter_alive_retry	Number of retries for monitoring connection to the Mirroring Controller arbitration process (number of times)	Specify the number of heartbeat retries to be performed if an error is detected in the heartbeat to the Mirroring Controller arbitration process. If the heartbeat does not respond within the specified number of retries, the Mirroring Controller arbitration process is determined to have been disconnected. Specify a value between 0 and 2147483647. The default is 0 times. This parameter does not need to be set for operation that does not use the arbitration server.
arbiter_connect_interval	Attempt interval for connection to the Mirroring Controller arbitration process (milliseconds)	Reconnection is attempted at the specified interval if connection fails at startup of the Mirroring Controller process or if the Mirroring Controller arbitration process is disconnected. Specify a value between 1 and 2147483647. The default is 16000 milliseconds. This parameter does not need to be set for operation that does not use the arbitration server.
arbiter_connect_timeout	Timeout for connection to the Mirroring Controller arbitration process (seconds)	If reconnection at startup of the Mirroring Controller process or after disconnection of the Mirroring Controller arbitration process does not succeed within the specified number of seconds, connection to the Mirroring Controller arbitration process is determined to have failed and reconnection is attempted. Specify a value between 1 and 2147483647. The default is 20 seconds. This parameter does not need to be set for operation that does not use the arbitration server.
fencing_command	'*fencingCmdFilePath*' [Setting example] fencing_command = 'c:\\mc\ \fencing_dir\\execute_fencing.bat'	Specify the full path of the fencing command that fences a database server where an error is determined to have occurred. Specify "\\" as the delimiter. Enclose the path in single quotation marks ('). Specify the path using less than 260 bytes. Any multibyte characters must use the same encoding as the operating system. This parameter must be specified when "command" is set for heartbeat_error_action.
fencing_command_timeout	Fencing command timeout (seconds)	If the command does not respond within the specified number of seconds, fencing is determined to have failed

Parameter	Value set	Explanation
		and a signal (SIGTERM) is sent to the fencing command execution process. Specify a value between 1 and 2147483647. The default is 20 seconds.
arbitration_timeout	Arbitration processing timeout in the Mirroring Controller arbitration process (seconds)	The specified value must be at least equal to the value of fencing_command_timeout in the arbitration configuration file, which is the heartbeat monitoring time of the operating system or server. If there is no response for at least the number of seconds specified, the primary server will not be switched and the standby server will not be disconnected. Therefore, perform degradation manually. If the heartbeat_interval, heartbeat_timeout, and heartbeat_retry values are specified in arbitration.conf for the arbitration server, use the arbitration server values to design arbitration_timeout. Specify a value between 1 and 2147483647. The default is 30 seconds. This parameter does not need to be set for operation that does not use the arbitration server.
arbitration_command	'*arbitrationCmdFilePath*' [Setting example] arbitration_command = 'c:\\mc\ \arbitration_dir\ \execute_arbitration_command.bat'	Specify the full path of the arbitration command to be executed when an abnormality is detected by heartbeat monitoring of the operating system or server. Specify "\\" as the delimiter. Enclose the path in single quotation marks ('). Any multibyte characters must use the same encoding as the operating system. Specify the path using less than 260 bytes. This parameter must be specified when "command" is set for heartbeat_error_action.
arbitration_command_timeout	Arbitration command timeout (seconds)	If the arbitration command does not respond within the specified number of seconds, it is determined that execution of the arbitration command has failed and a signal (SIGTERM) is sent to the arbitration command execution process. Specify a value between 1 and 2147483647. The default is 30 seconds. This parameter can be specified only when "command" is set for heartbeat_error_action.
shutdown_detached_synchronous_standby	on or off	Specify whether to forcibly stop the instance on the standby server when the standby server is disconnected. on: Stop the instance. off: Do not stop the instance. If "on" is specified and the pre-detach command was created, the pre-detach command is executed and then the instance is stopped.

Parameter	Value set	Explanation
		The default is "off".
post_switch_command	'*postSwtchCmdFilePath*' [Setting example] post_switch_command = 'c:\\mc\ \status_change\ \execute_post_switch.bat'	Specify the full path of the command to be called by Mirroring Controller after a new primary server is promoted during a failover of the primary server. Specify "\\" as the delimiter. Enclose the path in single quotation marks ('). Specify the path using less than 260 bytes.
post_attach_command	'*postAttachCmdFilePath*' [Setting example] post_attach_command = 'c:\\mc\ \status_change\ \execute_post_attach.bat'	Specify the full path of the command to be called by Mirroring Controller after the standby server is attached to the cluster system. Specify "\\" as the delimiter. Enclose the path in single quotation marks ('). Specify the path using less than 260 bytes.
pre_detach_command	'*preDetachCmdFilePath*' [Setting example] pre_detach_command = 'c:\\mc\ \status_change\ \execute_pre_detach.bat'	Specify the full path of the command to be called by Mirroring Controller before the standby server is disconnected from the cluster system. Specify "\\" as the delimiter. Enclose the path in single quotation marks ('). Specify the path using less than 260 bytes.
status_change_command_timeout	State transition command timeout (seconds)	Specify the timeout value of the post-switch command, post-attach command, and pre-detach command. If the command does not respond within the specified number of seconds, a signal (SIGTERM) is sent to the execution process of the status change command. Specify a timeout between 1 and 2147483647. The default is 20 seconds.
check_synchronous_standby_names_validation	on or off	Specify whether Mirroring Controller is to periodically check during operations whether the synchronous_standby_names parameter in postgresql.conf was changed by an incorrect user operation. However, it is not recommended to enable this parameter, because performing this check causes Mirroring Controller to use the CPU of the database server redundantly and execute SQL statements at high frequency. This parameter is compatible with operations in FUJITSU Enterprise Postgres 9.6 or earlier. The default is "off".
db_instance_ext_pq_conninfo	*'libpqConnectionSSLParmToConnectToDbinstance'*	Specify, in key-value form, the connection parameter for libpq that Mirroring Controller adds when connecting to a database. The connection parametersyou can specify are those related to SSL. Use ASCII characters to specify this parameter. The connection parameter specified in this parameter must also be specified in the db_instance_ext_jdbc_conninfo.
db_instance_ext_jdbc_conninfo	*'JDBCConnectionSSLParmToConnectToDbinstance'*	Specify, in URI form, the connection parameter for JDBC that Mirroring Controller adds when connecting to a

Parameter	Value set	Explanation
		database. The connection parametersyou can specify are those related to SSL. Use ASCII characters to specify this parameter. The connection parameter specified in this parameter must also be specified in the db_instance_ext_pq_conninfo.

*1: By using an event source name that is similar to the specified event_source parameter of the postgresql.conf file, the Mirroring Controller output content can be referenced transparently, so log reference is easy.

*2: The operating system TCP connection timeout period is determined by the TCP global parameters initial RTO and maximum SYN retransmits. The remote_call_timeout parameter must be set to a value that is shorter than the timeout period for the operating system TCP connection timeout, so change either parameter as necessary.

*3: In management communications, arbitrage processing, fencing commands, and state transition commands may be executed in succession by the Mirroring Controller arbitrage process. Therefore, the value specified for the remote_call_timeout parameter must be greater than the sum of these timeout values. Depending on the value specified for the heartbeat_error_action parameter, set the remote_call_timeout parameter using the following formula.

- arbitration: (arbitration_timeout + fencing_command_timeout + status_change_command_timeout) * 1000
- command: (fencing_command_timeout + status_change_command_timeout) * 1000
- message: (fencing_command_timeout + status_change_command_timeout) * 1000
- fallback: (status_change_command_timeout) * 1000

Because other internal processing may be performed in the management communication, set the value obtained by multiplying the calculation result of the above equation by the safety factor (about 1.2).

Also, for the fencing_command_timeout parameter, use the value of the parameter in the database server's server definition file, not in the arbitration definition file.

The availability of some parameters depends on the value set for the heartbeat_error_action parameter that sets the operation to be performed if heartbeat monitoring of the operating system or server detects a heartbeat abnormality.

Table A.5 Parameter availability depending on the value set for the heartbeat_error_action parameter

Parameter	Value set			
	arbitration	command	message	fallback
arbiter_alive_interval	Y	N	N	N
arbiter_alive_timeout	Y	N	N	N
arbiter_alive_retry	Y	N	N	N
arbiter_connect_interval	Y	N	N	N
arbiter_connect_timeout	Y	N	N	N
arbitration_timeout	Y	N	N	N
arbitration_command	N	R	N	N
arbitration_command_timeout	N	Y	N	N
fencing_command	Y	R	Y	N
fencing_command_timeout	Y	Y	Y	N
shutdown_detached_synchronous_standby	Y	Y	N	N

R: Required

Y: Can be specified

N: Cannot be specified

A.4.2 Arbitration Configuration File

In arbitration.conf, define the information related to arbitration and control of the Mirroring Controller arbitration process.

Linux

Table A.6 arbitration.conf file (Linux)

Parameter	Value set	Description
port	Port number of the Mirroring Controller arbitration process	The specified value must not exceed the range 0 to 65535. Ensure that the port number does not conflict with other software. Do not specify an ephemeral port that may temporarily be assigned by another program. For the port number of the arbitration server to be specified in network.conf on the database server, specify the same value as the port number specified in this parameter.
my_address	'*ipAddrOrHostNameThatAcceptsConnectionFromMirroringControllerProcessOnDbServer*' [Setting example] my_address = '192.0.3.120'	For the IP address or host name of the arbitration server to be specified in network.conf on the database server, specify the same value as the IP address or host name specified in this parameter. IPv4 and IPv6 addresses can be specified. Specify the IP address or host name, enclosed in single quotation marks (').
core_file_path	'*coreFileOutputDir*'	Specify the directory to which the core file is to be output, enclosed in single quotation marks ('). Use ASCII characters to specify this parameter. If this parameter is omitted, it will be assumed that the Mirroring Controller arbitration process management directory was specified.
syslog_facility	Specify LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, or LOCAL7.	When the import of logs to the syslog is enabled, the value of this parameter will be used for "facility" of the syslog. The default is "LOCAL0".
syslog_ident	'*programName*'	Specify using single quotation marks (') to enclose the program name used to identify the Mirroring Controller arbitration process message in the system log. Use ASCII characters excluding spaces to specify this parameter. The default is 'MirroringControllerArbiter'.
fencing_command	'*fencingCmdFilePath*' [Setting example] fencing_command = '/arbiter/fencing_dir/execute_fencing.sh'	Specify the full path of the fencing command that fences a database server where an error is determined to have occurred. Enclose the path in single quotation marks ('). Specify the path using less than 1024 bytes.
fencing_command_timeout	Fencing command timeout (seconds)	If the command does not respond within the specified number of seconds, fencing is determined to have failed and a signal (SIGTERM) is sent to the fencing command execution process. Specify a value between 1 and 2147483647.

Parameter	Value set	Description
		The default is 20 seconds.
heartbeat_interval(*1)	Interval time for heartbeat monitoring of the operating system or server (milliseconds)	The heartbeat monitoring of the database server is checked at the specified interval and arbitration is performed. Specify a value between 1 and 2147483647. The default is the value specified in *serverIdentifier*.conf of the database server. Specify this parameter to perform optimization taking into account differences in the line load to the admin network and the reduction in the time it takes to degrade.
heartbeat_timeout	Timeout for heartbeat monitoring of the operating system or server (seconds)	If there is no response for at least the number of seconds specified, it will be assumed that an error has occurred that requires the primary server or standby server to be fenced. Specify a value between 1 and 2147483647. The default is the value specified in *serverIdentifier*.conf of the database server. Specify this parameter to perform optimization taking into account differences in the line load to the admin network and the reduction in the time it takes to degrade.
heartbeat_retry	Number of retries for heartbeat monitoring of the operating system or server (number of times)	Specify the number of retries to be performed when an error has been detected that requires the primary server or standby server to be fenced. If an error is detected in succession more than the specified number of times, fencing will be performed. Specify a value between 0 and 2147483647. The default is the value specified in *serverIdentifier*.conf of the database server. Specify this parameter to perform optimization taking into account differences in the line load to the admin network and the reduction in the time it takes to degrade.

*1:Refer to "2.11.4 Tuning for Optimization of Degradation Using Abnormality Monitoring" for information on the tuning parameters for operating system or server abnormality monitoring when using an arbitration server.

Windows

Table A.7 arbitration.conf file (Windows)

Parameter	Value set	Description
port	Port number of the Mirroring Controller arbitration process	The specified value must not exceed the range 0 to 65535. Ensure that the port number does not conflict with other software. Do not specify an ephemeral port that may temporarily be assigned by another program. For the port number of the arbitration server to be specified in network.conf on the database server,

Parameter	Value set	Description
		specify the same value as the port number specified in this parameter.
my_address	'*ipAddrOrHostNameThatAcceptsConnectionFromMirroringControllerProcessOnDbServer*' [Setting example] my_address = '192.0.3.120'	For the IP address or host name of the arbitration server to be specified in network.conf on the database server, specify the same value as the IP address or host name specified in this parameter. IPv4 and IPv6 addresses can be specified. Specify the IP address or host name, enclosed in single quotation marks (').
core_file_path	'*coreFileOutputDir*'	Specify the directory to which the core file is to be output, enclosed in single quotation marks ('). Use ASCII characters, and specify "\\" as the path delimiter. If this parameter is omitted, it will be assumed that the Mirroring Controller arbitration process management directory was specified.
service_name	'*registeredServiceNameOfMirroringControllerArbitrationProcess*'	Specify the Mirroring Controller arbitration process service name to be registered as a Windows service, enclosed in single quotation marks ('). Use ASCII characters excluding forward slash (/) and backslash (\) to specify this parameter. The service name is up to 124 bytes. The default is 'MirroringControllerArbiter'.
event_source	*'eventSourceName'*	Specify the event source name to be used to identify the Mirroring Controller arbitration process message in the event log, enclosed in single quotation marks ('). Use ASCII characters to specify this parameter. The maximum length of the event source name is 255 bytes. The default is 'MirroringControllerArbiter'.
fencing_command	'*fencingCmdFilePath*' [Setting example] fencing_command = 'c:\\arbiter\ \fencing_dir\\execute_fencing.bat'	Specify the full path of the fencing command that fences a database server where an error is determined to have occurred. Specify "\\" as the delimiter. Enclose the path in single quotation marks ('). Specify the path using less than 260 bytes. Any multibyte characters must use the same encoding as the operating system.
fencing_command_timeout	Fencing command timeout (seconds)	If the command does not respond within the specified number of seconds, fencing is determined to have failed and a signal (SIGTERM) is sent to the fencing command execution process. Specify a value between 1 and 2147483647. The default is 20 seconds.

Parameter	Value set	Description
heartbeat_interval(*1)	Interval time for heartbeat monitoring of the operating system or server (milliseconds)	The heartbeat monitoring of the database server is checked at the specified interval and arbitration is performed. Specify a value between 1 and 2147483647. The default is the value specified in *serverIdentifier*.conf of the database server. Specify this parameter to perform optimization taking into account differences in the line load to the admin network and the reduction in the time it takes to degrade.
heartbeat_timeout	Timeout for heartbeat monitoring of the operating system or server (seconds)	If there is no response for at least the number of seconds specified, it will be assumed that an error has occurred that requires the primary server or standby server to be fenced. Specify a value between 1 and 2147483647. The default is the value specified in *serverIdentifier*.conf of the database server. Specify this parameter to perform optimization taking into account differences in the line load to the admin network and the reduction in the time it takes to degrade.
heartbeat_retry	Number of retries for heartbeat monitoring of the operating system or server (number of times)	Specify the number of retries to be performed when an error has been detected that requires the primary server or standby server to be fenced. If an error is detected in succession more than the specified number of times, fencing will be performed. Specify a value between 0 and 2147483647. The default is the value specified in *serverIdentifier*.conf of the database server. Specify this parameter to perform optimization taking into account differences in the line load to the admin network and the reduction in the time it takes to degrade.

*1:Refer to "2.11.4 Tuning for Optimization of Degradation Using Abnormality Monitoring" for information on the tuning parameters for operating system or server abnormality monitoring when using an arbitration server.

Appendix B Supplementary Information on Building the Primary Server and Standby Server on the Same Server

The primary server and standby server can be pseudo-configured on the same server for system testing, for example. Out of consideration for performance and reliability, do not use this type of configuration for any other purposes. For this reason, do not use this type of configuration in a production environment.

Note that the setup and operations is the same as if the primary and standby servers are built on different servers.

This appendix provides supplementary information explaining how to configure the primary server and standby server on the same server.

Note

Even if automatic degradation by an arbitration server is set when the primary server and standby server are configured on the same server, there will be no effect of it.

B.1 Backup Data Storage Destination Directory

It is not a problem if the same backup data storage destination directory is used on the primary server and standby server.

B.2 Registering Service Names and Event Source Names in the Windows Service

Ensure that the following names of resources to be registered on the operating system by Fujitsu Enterprise Postgres and Mirroring Controller are not duplicated between the primary server and standby server:

- Service name registered in the Windows service
- Event source name

B.3 How to Execute the mc_ctl Command

When executing the mc_ctl command, specify the server identifier in the --local-server option in order to identify the operation destination server.

Below is an example of starting Mirroring Controller of the server "server1" defined in the network.conf file. For mc_ctl command operations using another mode, also specify the --local-server option.

Define two server identifiers for the same IP address with different port numbers in the network.conf file.

```
Example)
```

```
server1 192.0.2.100 27540
server2 192.0.2.100 27541
```

Ensure that the port numbers of both primary server and standby server do not conflict with any other software.

Enabling automatic switch/disconnection

Start Mirroring Controller of the server "server1":

```
Example)
```

```
> mc_ctl start -M D:\mcdir\inst1 --local-server server1
```

Stop Mirroring Controller of the server "server1":

Example)

```
> mc_ctl stop -M D:\mcdir\inst1 --local-server server1
```

Disabling automatic switch/disconnection

Start Mirroring Controller of the server "server1":

Example)

```
> mc_ctl start -M D:\mcdir\inst1 -F --local-server server1
```

Stop Mirroring Controller of the server "server1":

Example)

```
> mc_ctl stop -M D:\mcdir\inst1 --local-server server1
```

Note

To specify the mc_ctl command with register mode, for registering to the Windows service, and the mc_ctl command with unregister mode, for unregistering from the Windows service, add the --local-server option.

Appendix C User Commands

This appendix describes three categories of commands:

- Fencing command
- Arbitration command
- State transition commands

This appendix describes each category of user command.

C.1 Fencing Command

Format

The syntax for calling the fencing command from the Mirroring Controller process or the Mirroring Controller arbitration process is described below.

Fencing command of the database server

```
fencingCmd executionMode mcDegradationOper cmdServerId targetServerId primarycenter
```

Fencing command of the arbitration server

```
fencingCmd executionMode mcDegradationOper targetServerId
```

Input

Fencing command of the database server

Execution mode

monitor: Detect issues via automatic monitoring of the Mirroring Controller process

command: Mirroring Controller command execution (switch mode or detach mode of the mc_ctl command)

Degradation operation to be performed by Mirroring Controller

switch: Switch

detach: Disconnect

cmdServerId

ID of the database server that called the command

targetServerId

ID of the database server to be fenced

primarycenter

Fixed value

Fencing command of the arbitration server

Execution mode

monitor: Detect issues via automatic monitoring of the Mirroring Controller process

command: Mirroring Controller command execution (switch mode or detach mode of the mc_ctl command)

Degradation operation to be performed by Mirroring Controller

switch: Switch

detach: Disconnect

targetServerId

ID of the database server to be fenced

Output

Return value

0: Mirroring Controller will continue the degradation process.

Other than 0: Mirroring Controller will cancel the degradation process.

Description

Identifies the database server targeted for fencing based on the input server identifier, and implements the process that isolates it from the cluster system.

Notes

- The command is executed by the operating system user who started Mirroring Controller or the Mirroring Controller arbitration process. Therefore, if the command is to be executed by a specific operating system user, change the executing user of the command accordingly.
- The operating system user who started Mirroring Controller or the Mirroring Controller arbitration process must have execution privileges to the command. Otherwise, the degradation process will be canceled.
- From a security point of view, set the access privileges as necessary so that the fencing command is not overwritten and unauthorized operations are not performed by unintended operating system users.
- If the fencing command returns a value other than 0, Mirroring Controller will cancel the degradation process, so it is necessary for the user to check the status of the server, and switch or disconnect it manually.
- Before executing the fencing command, check if the server is already fenced, to avoid the command terminating abnormally.
- If the command times out, Mirroring Controller will stop the command, output an error message, and cancel the degradation process.

Information

The fencing command can be implemented by simply stopping the operating system or server. For example, if stopping the power for the database server, it is possible to use a utility to control the hardware control board in environments equipped with boards compatible with IPMI hardware standard.

Linux

Below is a sample script of a fencing command that powers off the database server using the IPMI tool.

Sample shell script

```
/installDir/share/mcarb_execute_fencing.sh.sample
```

Windows

Below is a sample script of a fencing command that powers off the database server using IPMIUTIL.

Sample shell script

```
installDir\share\mcarb_execute_fencing.bat.sample
```

C.2 Arbitration Command

Format

The syntax for calling the arbitration command from the Mirroring Controller process is described below.

```
arbitrationCmd cmdServerId targetServerId primarycenter
```

Input

cmdServerId

ID of the database server that called the command

targetServerId

ID of the database server to arbitrate

primarycenter

Fixed value

Output

Return value

0: The database server to arbitrate has an issue, and Mirroring Controller will continue the degradation process.

Other than 0: The database server to arbitrate is normal, and Mirroring Controller will cancel the degradation process.

Description

Identifies the database server to arbitrate based on the input server identifier, and checks the status of the server.

Notes

- The command is executed by the operating system user who started Mirroring Controller.
- The operating system user who started Mirroring Controller must have execution privileges to the command. Otherwise, the command will not be called, and the degradation process will be canceled.
- If the command times out, Mirroring Controller will stop the command, output an error message, and cancel the degradation process.

C.3 State Transition Commands

State transition commands include the three types of user commands below. Any of the commands can be implemented by Mirroring Controller in conjunction with database server status transitions.

- Post-switch command
- Pre-detach command
- Post-attach command

C.3.1 Post-switch Command

Format

The syntax for calling the post-switch command from the Mirroring Controller process is described below.

```
postswitchCmd serverIdentifier primarycenter
```

Input

serverIdentifier

ID of the database server (new primary server) that was switched

primarycenter

Fixed value

Output

Return value

None

Notes

- The command is executed by the operating system user who started Mirroring Controller.
- The operating system user who started Mirroring Controller must have execution privileges to the command. Otherwise, the command will not be called.
- If the command times out, Mirroring Controller will stop the command, output an error message, and cancel the process.

C.3.2 Pre-detach Command

Format

The syntax for calling the pre-detach command from the Mirroring Controller process is described below.

```
predetachCmd cmdServerId serverRole targetServerId primarycenter
```

Input

cmdServerId

ID of the database server that called the command

Server role

Role of the database server that called the command

primary: Primary

standby: Standby

targetServerId

ID of the standby server to be disconnected from the cluster system

primarycenter

Fixed value

Output

Return value

None

Notes

- The command is executed by the operating system user who started Mirroring Controller.
- The operating system user who started Mirroring Controller must have execution privileges to the command. Otherwise, the command will not be called, however, Mirroring Controller will output an error message and continue the process.
- If the command times out, Mirroring Controller will stop the command, output an error message, and cancel the process.

C.3.3 Post-attach Command

Format

The syntax for calling the post-attach command from the Mirroring Controller process is described below.

```
postattachCmd cmdServerId serverRole targetServerId primarycenter
```

Input

cmdServerId

ID of the database server that called the command

Server role

Role of the database server that called the command

primary: Primary

standby: Standby

targetServerId

ID of the standby server to be attached to the cluster system

primarycenter

Fixed value

Output

Return value

None

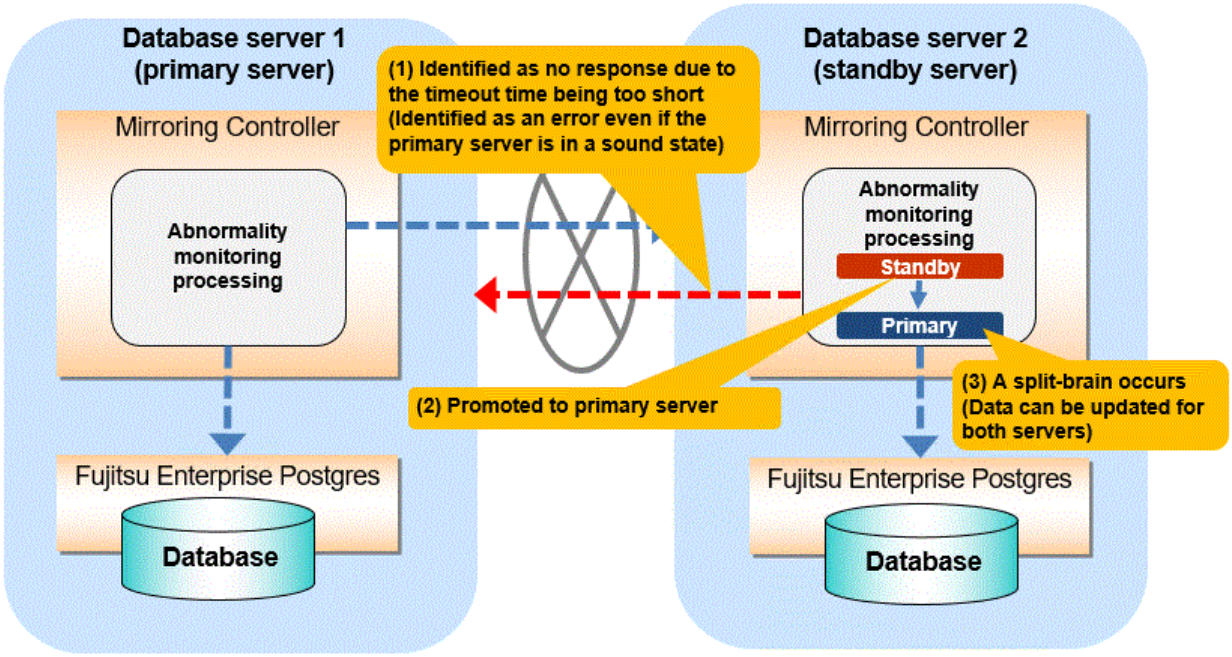
Notes

- The command is executed by the operating system user who started Mirroring Controller.
- The operating system user who started Mirroring Controller must have execution privileges to the command. Otherwise, the command will not be called.
- If the command times out, Mirroring Controller will stop the command, output an error message, and cancel the process.

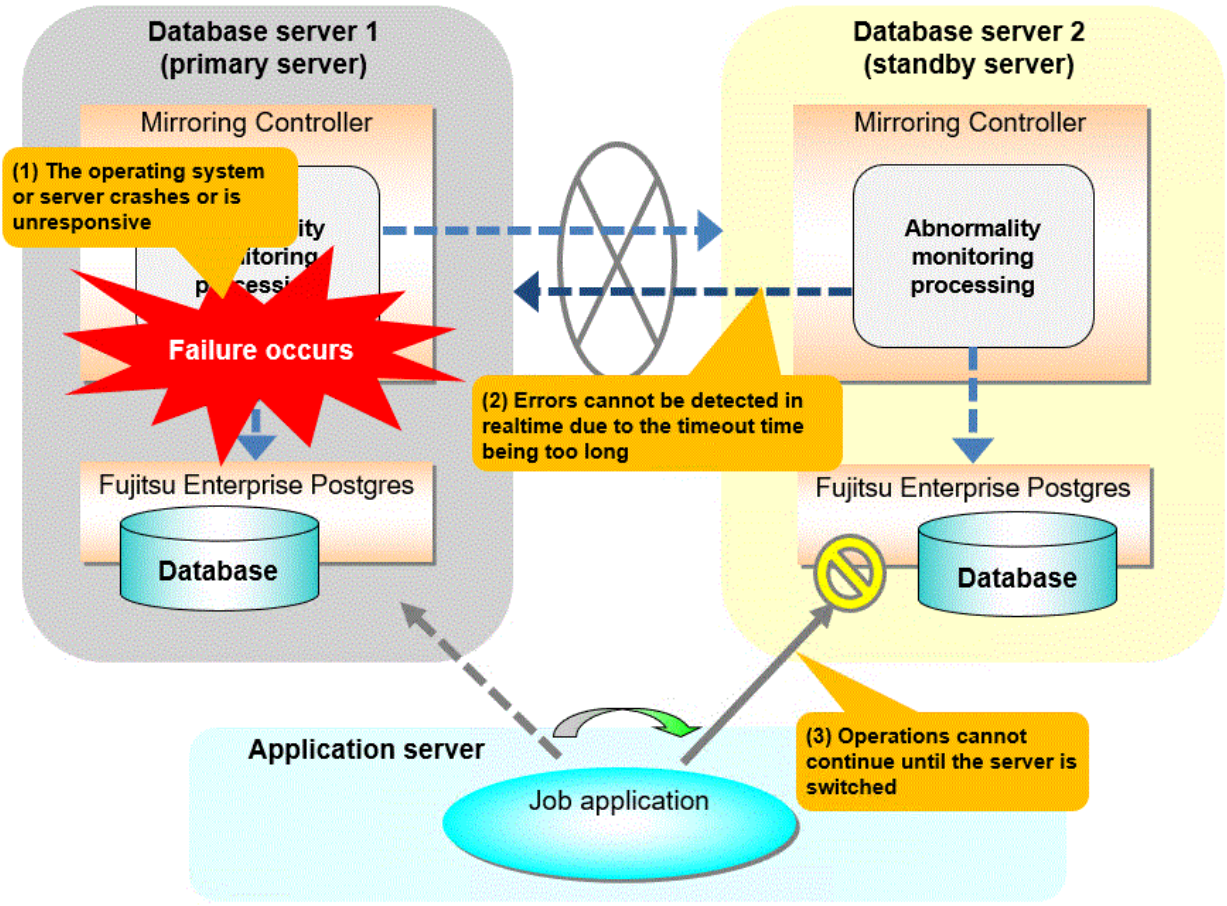
Appendix D Notes on Performing Automatic Degradation Immediately after a Heartbeat Abnormality

The type of issue below occurs if automatic degradation is performed unconditionally after an issue is detected during heartbeat monitoring of an operating system or server, and heartbeat monitoring was not properly tuned.

- **If the timeout time is too short**



● If the timeout time is too long



Notes on monitoring when the operating system or server crashes or is unresponsive

As illustrated in the diagram above, timeout is used to monitor whether the operating system or server crashes or is unresponsive. Therefore, if tuning has not been performed correctly, there is a risk of a split-brain mistakenly occurring even if the server is in a sound state.

Split-brain is a phenomenon in which both servers temporarily operate as primary servers, causing data updates to be performed on both servers.

Split-brain detection method

It can be confirmed that split-brain occurs under the following conditions:

1. When the mc_ctl command is executed in status mode on both servers, the "host_role" of both servers is output as "primary", and
2. The following message is output to the system log of one of the servers:

```
promotion processing completed (MCA00062)
```

How to recover from a split-brain

Use the procedure described below. Note that the new primary server is the server that was confirmed in step 2 of the aforementioned detection method.

1. Stop all applications that are running on the old and new primary servers.
2. Investigate and recover the database.
   Investigate the update results that have not been reflected to the new primary server from the database of the old primary server, and apply to the new primary server as necessary.
3. Stop the old primary server instance and the Mirroring Controller.
4. Resume the applications that were stopped in step 1.

5. Recover the old primary server.
   While referring to "2.5 Setting Up the Standby Server", build (set up) the old primary server as the new standby server, from the new primary server.

Appendix E Supplementary Procedure on Configuring for Operation in Database Multiplexing Mode

This appendix explains a supplementary procedure on the configuration required for operation in database multiplexing mode.

E.1 Security Policy Settings

This section explains how to configure the security settings to enable an operating system user account designated as an instance administrator user to log on as a service.

1. Displaying the [Local Security Policy] window

   In Windows, select [Administrative Tools], and then click [Local Security Policy].

2. Setting up security

   1. In the [Local Security Policy] window, select [Security Settings], select [Local Policies], and then click [User Rights Assignment].
   2. Under [Policy] in the [User Rights Assignment] window, double-click [Log on as a service].
   3. In the [Log on as a service Properties] window, set the following:
      a. Select the [Local Security Setting] tab.
      b. On the [Local Security Setting] tab, click [Add User or Group].
      c. In the [Select Users or Groups] window, enter the operating system user account of the instance administrator user in [Enter the object names to select].
      d. Click [OK].
   4. In the [Log on as a service Properties] window, click [OK].
   5. From the [Local Security Policy] tree, click [Local Policies], and then double-click [Security Options].
   6. Scroll down and double-click [User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode].
   7. From the drop-down menu, select the "Elevate without prompting" in the [Local Security Setting] tab.
   8. Click [OK].

E.2 Windows Firewall Settings

This section explains how to enable the port number used by Mirroring Controller, if the Windows firewall feature is enabled.

Windows Server(R) 2016:

1. In the [Windows Firewall] window, click [Advanced settings] on the left side of the window.
2. In the [Windows Firewall with Advanced Security] window, click [Inbound Rules] on the left side of the window.
3. Click [New Rule] on the right side of the window.
4. In the [New Inbound Rule Wizard] window, select [Port], and then click [Next].
5. Select [TCP] and [Specific local ports], then specify the port number defined in the network definition file, and then click [Next].
6. Select [Allow the connection], and then click [Next].
7. Select the profiles for which this rule applies, and then click [Next].
8. In [Name], specify the desired name, and then click [Finish].
9. In the [Windows Firewall with Advanced Security] window, check if the added rule is enabled under [Inbound Rules] in the center of the window.

Other than above:

1. In the [Windows Defender Firewall] window, click [Advanced settings] on the left side of the window.
2. In the [Windows Defender Firewall with Advanced Security] window, click [Inbound Rules] on the left side of the window.
3. Click [New Rule] on the right side of the window.
4. In the [New Inbound Rule Wizard] window, select [Port], and then click [Next].
5. Select [TCP] and [Specific local ports], then specify the port number defined in the network definition file, and then click [Next].
6. Select [Allow the connection], and then click [Next].
7. Select the profiles for which this rule applies, and then click [Next].
8. In [Name], specify the desired name, and then click [Finish].
9. In the [Windows Defender Firewall with Advanced Security] window, check if the added rule is enabled under [Inbound Rules] in the center of the window.

Appendix F WebAdmin Disallow User Inputs Containing Hazardous Characters

WebAdmin considers the following as hazardous characters, which are not allowed in user inputs.

```
| (pipe sign)

& (ampersand sign)

; (semicolon sign)

$ (dollar sign)

% (percent sign)

@ (at sign)

' (single apostrophe)

" (quotation mark)

\' (backslash-escaped apostrophe)

\" (backslash-escaped quotation mark)

<> (triangular parenthesis)

() (parenthesis)

+ (plus sign)

CR (Carriage return, ASCII 0x0d)

LF (Line feed, ASCII 0x0a)

, (comma sign)

\ (backslash)
```

Appendix G Collecting Failure Investigation Data

If the cause of an error that occurs while building the environment or during operations is unclear, data must be collected for initial investigation.

This appendix describes how to collect data for initial investigation.

Using a database server to collect data

Use FJQSS (Information Collection Tool) to collect data for initial investigation.

Using an arbitration server to collect data

- Linux
  - RHEL7,RHEL8,RHEL9 and SLES 12

    Use FJQSS (Information Collection Tool) to collect data for initial investigation.
  - SLES 15

    Use the pgx_fjqssinf command to collect data for initial investigation.
- Windows

  Use FJQSS (Information Collection Tool) to collect data for initial investigation.

See

- Linux
  - Refer to the FJQSS manual for information on how to use FJQSS.
  - Refer to the Reference for information on the pgx_fjqssinf command.
- Windows

  Refer to the following manual for information on how to use FJQSS.
  - In the [Apps] menu, select [FJQSS (Information Collection Tool)], and then click [FJQSS User's Guide].

Note

Using a database server to collect data

- When using FJQSS to collect data for initial investigation, a window will be displayed for you to set the following environment variables:
  - Environment variable required for collecting data from the database

    Refer to "Collecting Failure Investigation Data" in the Operation Guide.
  - MCCONTROLDIR

    Set the Mirroring Controller management directory.
    Refer to "1.2.1 Mirroring Controller Resources" for information on the Mirroring Controller management directory.
- FJQSS operations must be performed by the instance administrator user.

Using an arbitration server to collect data

Linux

- When using FJQSS to collect data for initial investigation, you must set the following environment variables:
  - ARBCONTROLDIR

    Set the Mirroring Controller arbitration process management directory.
    Refer to "1.2.1 Mirroring Controller Resources" for information on the Mirroring Controller arbitration server management directory.
  - ARBUSER

    Specify the operating system user who started the Mirroring Controller arbitration process.
  - FSEP_SA_HOME

    Set the Server Assistant installation directory.

Windows

- When using FJQSS to collect data for initial investigation, a window will be displayed for you to set the following environment variables:
  - ARBCONTROLDIR

    Set the Mirroring Controller arbitration process management directory.
    Refer to "1.2.1 Mirroring Controller Resources" for information on the Mirroring Controller arbitration server management directory.

Information

FJQSS corresponds the required operating system of Fujitsu Enterprise Postgres described below.

- "Required Operating System" in the Installation and Setup Guide for Server.
- "Required Operating System" in the Installation and Setup Guide for Server Assistant.

Index

[A]

Action Required when a Heartbeat Abnormality is Detected.. 70
Action Required when All Database Servers or Instances Stopped 98
Action Required when an Error Occurs in the Database Multiplexing Mode..... 88
Action Required when Automatic Disconnection Fails..... 97
Action Required when Automatic Switch Fails..... 97
Action Required when Server Degradation Occurs..... 88
Addressing Errors During Degrading Operation..... 96
Application Connection Server Settings..... 45
arbitration.conf file (Linux)..... 129
arbitration.conf file (Windows)..... 130
Arbitration Command..... 136
Arbitration Configuration File..... 129
Arbitration Server Maintenance..... 79
Arbitration Server Process..... 10
Arbitration Server Resources..... 10
Authentication of the Standby Server..... 14

[B]

Backing up Database Multiplexing Mode Information..... 62
Backup Data Storage Destination Directory..... 133
Backup Operation..... 62

[C]

Changes in Operation..... 80
Changes Required when the Standby Server is Stopped..... 80
Changing from Database Multiplexing Mode to Single Server Mode..... 82
Changing from Single Server Mode to Database Multiplexing Mode..... 81
Changing Parameters..... 86
Changing to Database Multiplexing Mode when the Arbitration Server is Used for Automatic Degradation..... 85
Checking the Behavior..... 45
Checking the Connection Status..... 43
Checking the Connection Status on a Database Server..... 43
Checking the Connection Status on the Arbitration Server..... 44
Checking the Database Multiplexing Mode Status..... 68
Checking the Status of the Arbitration Server..... 69
Checking the Status of the Database Server..... 68
Configuring ICMP..... 16
Configuring the Arbitration Server..... 20
Confirming the Streaming Replication Status..... 42
Creating, Setting, and Registering the Primary Server Instance 31
Creating, Setting, and Registering the Standby Server Instance 38
Creating Applications..... 45

[D]

Database Backup Operation..... 62
Database Server Processes..... 10
Deciding on Operation when a Heartbeat Abnormality is Detected..... 11

[E]

Encryption of Transaction Logs Transferred to the Standby Server..... 15

[F]

Failback of the Primary Server..... 93
Fencing Command..... 135

[H]

How to Execute the mc_ctl Command..... 133

[I]

Identify cause of error and perform recovery..... 91,96
Identify Cause of Error and Restore the Standby Server..... 90
Identify Cause of Error and Restructure the Standby Server... 95
If Performing the Referencing Job on the Synchronous Standby Server..... 6
If Prioritizing the Main Job on the Primary Server..... 6
Installation..... 17

[M]

Manually Disconnecting the Standby Server..... 70
Manually Switching the Primary Server..... 69
Matching the system times..... 16
Mirroring Controller Resources..... 9
Monitoring Mirroring Controller Messages..... 71
Monitoring Using Database Multiplexing Mode..... 4

[N]

network.conf file..... 118
Network Configuration File..... 117
Notes on CPU Architecture and Products..... 11
Notes on Performing Automatic Degradation Immediately after a Heartbeat Abnormality..... 140

[O]

Operations in Database Multiplexing Mode..... 64
Operations when the Server has Started Degrading after a Disconnection has Occurred..... 94
Operations when the Server has Started Degrading after a Switch has Occurred..... 88
Overview of Database Multiplexing Mode..... 1

[P]

Parameters..... 113
Parameters Set on the Primary Server..... 113
Parameters Set on the Standby Server..... 115
Post-attach Command..... 138
Post-switch Command..... 137
postgresql.conf file..... 113,115
Pre-detach Command..... 138
Preparatory Tasks for the Output of Error Logs to the Event Log 18
Preparing for Setup..... 18
Preparing the Arbitration Server..... 19
Preparing the Backup Disk..... 18
Preparing the Database Server..... 18

Preparing to Output Error Logs to the Event Log (Windows). 19

[R]

Rebuild the Standby Server..... 93,96
Recovering from an Incorrect User Operation..... 101
Recovery of the Mirroring Controller management directory 91,96
Redundancy of the Admin and Log Transfer Networks.....11
Referencing on the Standby Server..... 6
Registering Service Names and Event Source Names in the Windows Service.....133
Rolling Updates..... 73

[S]

Security in Database Multiplexing..... 12
Security Policy Settings.....19,143
Security Policy Settings (Windows).....20
ServerConfiguration File..... 120
Server Configuration File for the Database Servers.....120
serverIdentifier.conf file..... 121
Server Maintenance..... 73
Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances.....59
Setting Automatic Start and Stop of the Mirroring Controller Arbitration Process..... 60
Setting Up Database Multiplexing Mode..... 16
Setting Up Database Multiplexing Mode on the Primary Server26
Setting Up Database Multiplexing Mode on the Standby Server36
Setting Up the Arbitration Server.....20
Setting Up the Primary Server.....26
Setting Up the Standby Server.....36
Setup..... 16
Starting and Stopping Mirroring Controller..... 65
Starting and Stopping the Mirroring Controller Arbitration Process..... 64
Starting Mirroring Controller..... 65
Starting Mirroring Controller on the Primary Server..... 35
Starting Mirroring Controller on the Standby Server.....40
Starting the Mirroring Controller Arbitration Process..... 25,64
State Transition Commands.....137
Stop Mirroring Controller.....90,95
Stopping for Maintenance..... 78
Stopping Mirroring Controller.....67
Stopping the Mirroring Controller Arbitration Process.....64
Supplementary Information on Building the Primary Server and Standby Server on the Same Server..... 133
Supplementary Procedure on Configuring for Operation in Database Multiplexing Mode..... 143
System Configuration for Database Multiplexing Mode..... 6

[T]

Tuning.....45
Tuning for Optimization of Degrading Operation Using Abnormality Monitoring.....46
Tuning to Stabilize Queries on the Standby Server.....45
Tuning to Stabilize Queries on the Standby Server (when Performing Frequent Updates on the Primary Server)..... 46
Tuning to Stabilize the Database Multiplexing Mode.....45

[U]

Uninstalling in Database Multiplexing Mode..... 86
Users who perform setup and operations on the arbitration server 16
Users who perform setup and operations on the database server16

[W]

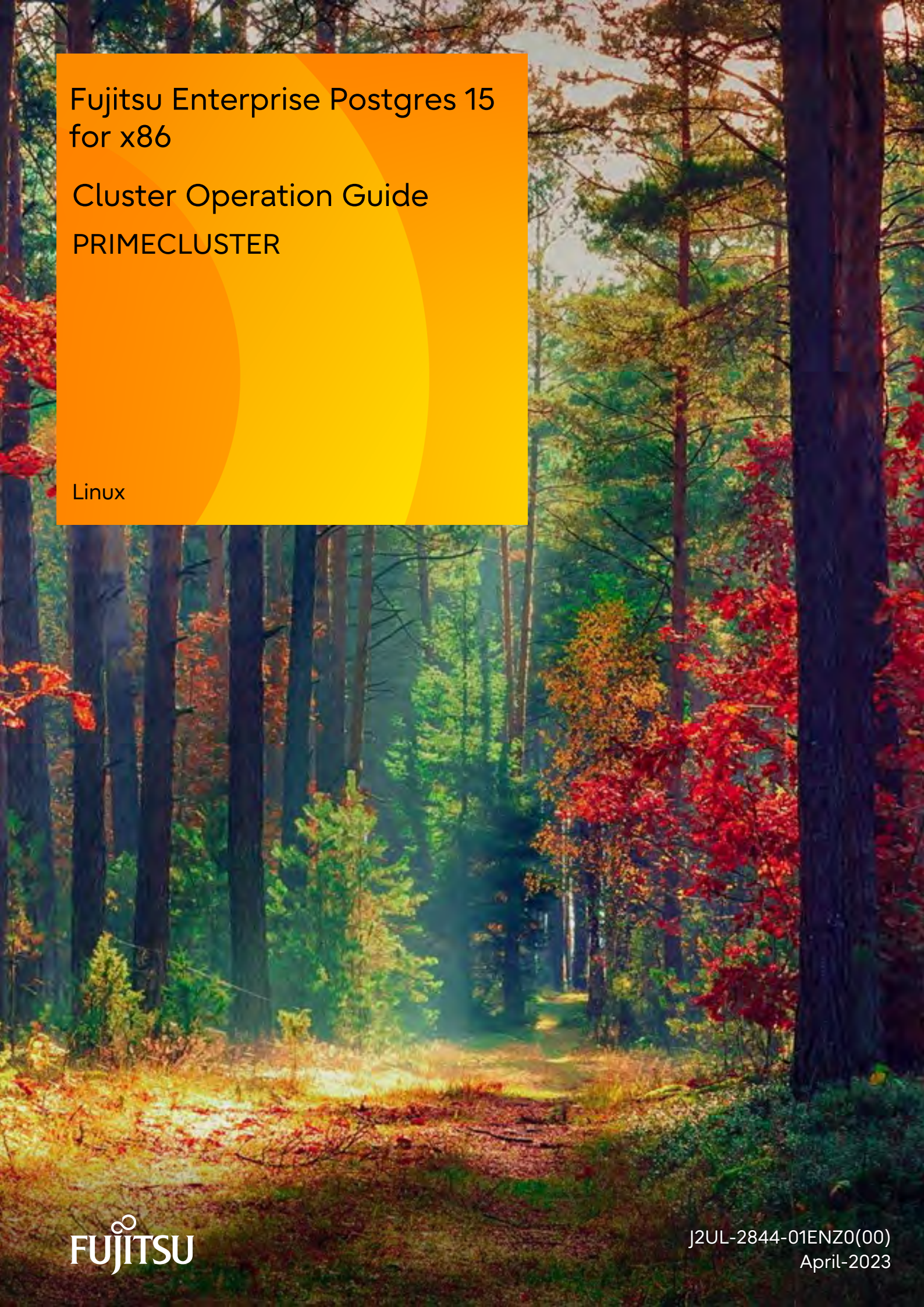
What is Database Multiplexing Mode.....1
Windows Firewall Settings.....143

Fujitsu Enterprise Postgres 15 for x86

Cluster Operation Guide

PRIMECLUSTER

Linux



FUJITSU

J2UL-2844-01ENZ0(00)
April-2023

Preface

Purpose of this document

This document explains the items required to operate Fujitsu Enterprise Postgres on a cluster system that uses PRIMECLUSTER.

Intended readers

This document is aimed at people who install and operate Fujitsu Enterprise Postgres on a cluster system that uses PRIMECLUSTER. Readers of this document are also assumed to have general knowledge of:

- PRIMECLUSTER
- PostgreSQL
- SQL
- Linux

Structure of this document

This document is structured as follows:

Chapter 1 Overview of Failover Operation

Provides an overview of failover operation.

Chapter 2 Setting Up Failover Operation

Explains how to set up failover operation.

Chapter 3 Failover Operation

Explains the procedures involved in failover operation.

Chapter 4 Procedures Required after a Failover Error

Explains the procedures required after an abnormality occurs during failover operation.

Appendix A Creating Resources and Creating/Modifying Cluster Applications

Explains how to create and modify cluster applications that include Fujitsu Enterprise Postgres resources.

Appendix B Command Reference

Explains for details in command.

Appendix C Failover Operation in Streaming Replication

Explains failover operations in streaming replication.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version


```
Edition 1.0: April 2023
```


Copyright

Copyright 2018-2023 Fujitsu Limited

Contents

Chapter 1 Overview of Failover Operation 1
1.1 Definition of Failover Operation 1
1.2 System Configuration of Failover Operation 2

Chapter 2 Setting Up Failover Operation 5
2.1 Installing PRIMECLUSTER and Fujitsu Enterprise Postgres 6
2.2 Configuring PRIMECLUSTER 6
2.3 Creating a GDS Volume 6
2.4 Creating a File System 6
2.4.1 Creating a File System 7
2.4.2 Editing Configuration Files 7
2.4.3 Checking 7
2.5 Creating an Operating System User to Start Fujitsu Enterprise Postgres 7
2.6 Mounting the File System 8
2.7 Creating a Database Cluster 8
2.8 Registering Resource Information for the Fujitsu Enterprise Postgres Database Cluster 9
2.9 Configuring Storage Data Protection Using Transparent Data Encryption 11
2.10 Unmounting the File System 12
2.11 Creating a Cluster Application 12
2.12 Application Development 12
2.12.1 Precautions when Developing Applications 12
2.12.2 Corrective Action with Application when State Transition Occurs 12
2.13 Checking Operation 15

Chapter 3 Failover Operation 16
3.1 Adding Tablespaces 16
3.2 Modifying Database Cluster Resources 16
3.3 Operations under Normal Circumstances 17
3.4 Operation at State Transition 17
3.5 Maintenance Tasks 18
3.5.1 Simultaneous Stopped Node Maintenance 19
3.5.1.1 Stopping the Cluster Applications on Both Nodes for Node Maintenance 19
3.5.1.2 Maintenance Tasks on Both Nodes 20
3.5.1.3 Restarting the Cluster Applications Stopped on Both Nodes for Node Maintenance 20
3.5.2 Mutual Switch Maintenance 20
3.5.2.1 Stopping the Cluster Applications on the Standby Node for Node Maintenance 20
3.5.2.2 Standby Node Maintenance 21
3.5.2.3 Restarting the Cluster Applications Stopped on the Standby Node for Node Maintenance 21
3.5.2.4 Mutual Switch 21
3.5.2.5 Standby Node Maintenance after the Mutual Switch 21

Chapter 4 Procedures Required after a Failover Error 22
4.1 Identifying the Cause of an Error 22
4.2 Stopping the Cluster Applications for Node Maintenance 23
4.3 Starting the Database Cluster 23
4.4 Recovery Tasks 23
4.5 Stopping the Database Cluster 23
4.6 Restarting the Cluster Applications Stopped for Node Maintenance 23

Appendix A Creating Resources and Creating/Modifying Cluster Applications 24
A.1 Creating Resources and Cluster Applications 24
A.1.1 Preparing for Creation 24
A.1.2 Creating a Cluster Application and Configuring Its Attributes 24
A.1.3 Creating Gds Resources 31
A.1.4 Creating Fujitsu Enterprise Postgres Resources 32
A.1.5 Creating Gls or Takeover Network Resources 33

A.1.6 Creating Fsystem Resources....34
A.1.7 Creating Resources and Finalizing Cluster Application Creation....35
A.2 Modifying Cluster Applications....36
A.2.1 Moving to the Cluster Application Edit Window....36
A.2.2 Adding a Created Shared Disk Class to the Cluster Application....37
A.2.3 Adding a Created File System to the Cluster Application....38
A.2.4 Reflecting Modifications Made to the Cluster Application....38

Appendix B Command Reference....40
B.1 pgx_pclrsc....40

Appendix C Failover Operation in Streaming Replication....42
C.1 Possible Cluster Operation....42
C.2 Overview of Failover Operation....42
C.2.1 Supported Operational Forms and System Configurations....42
C.3 Set up failover operation....44
C.3.1 Primary Server Setup....45
C.3.2 Standby Server Setup....47
C.4 Failover Operation....52
C.5 Procedures Required after a Failover Error....53
C.5.1 In Case of Primary Server Error....53
C.5.2 In Case of an Error on the Standby Server....55

Index....56

Chapter 1 Overview of Failover Operation

This chapter provides an overview of failover operation in Fujitsu Enterprise Postgres.

1.1 Definition of Failover Operation

When an abnormality occurs on any of the multiple server devices that make up the cluster system (hereafter referred to as "nodes"), the failover feature transfers a job operating on such a node to another one. Failover can reduce the time for which jobs are stopped when an abnormality occurs, and allows jobs to continue while the node on which the abnormality occurred is recovered.

In Fujitsu Enterprise Postgres, failover can be integrated with PRIMECLUSTER. In such a configuration, the shared disk (GDS) provided by PRIMECLUSTER is used. A server in the cluster system receives the application processing, acting as the active server (active node).

For example, if the active node fails, another server (standby node) inherits the shared disk content and becomes the active node. Using the PRIMECLUSTER feature, the applications to be run on the active node for the database can also be linked with the database and switched. Therefore, jobs can be continued even during recovery of a node where an abnormality has occurred. There is also no need to send the updated content of the database on the active node to the standby node, so the processing performance will be the same as that of a non-cluster system.

Note

When failover operation is integrated with PRIMECLUSTER, the disk is not referenced by the standby node, and therefore failover operation cannot be used for load distribution.

In addition, failover operation cannot be performed concurrently with database multiplexing.

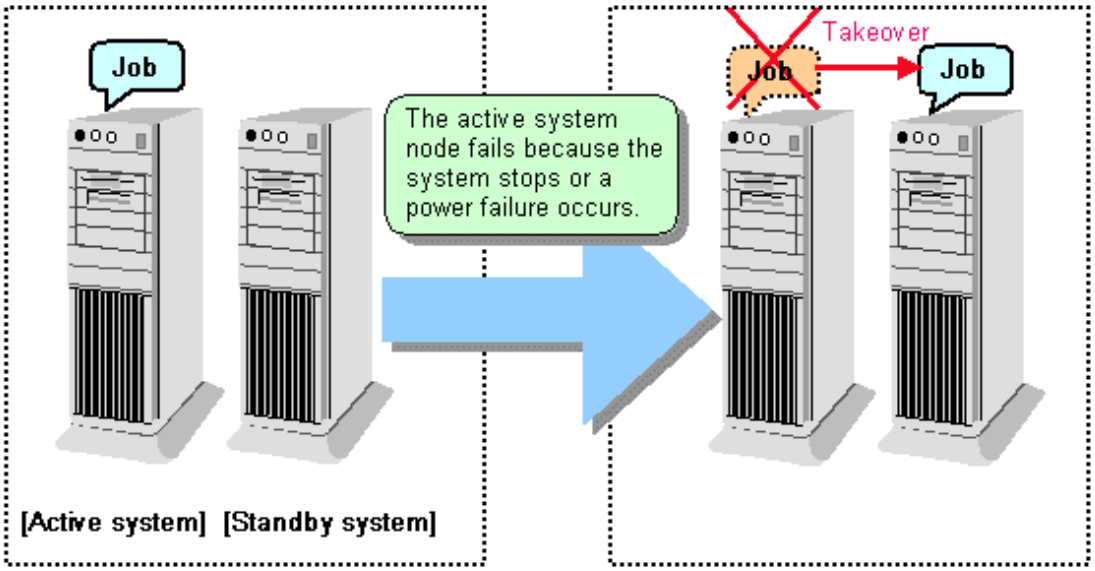
Information

Streaming replication, which enables load balancing, can be combined with failover operations.

For details, see "Appendix C Failover Operation in Streaming Replication".

Hereinafter, failover operation integrated with PRIMECLUSTER is referred to as "failover operation".

Figure 1.1 Concept of failover



Feature of failover operation

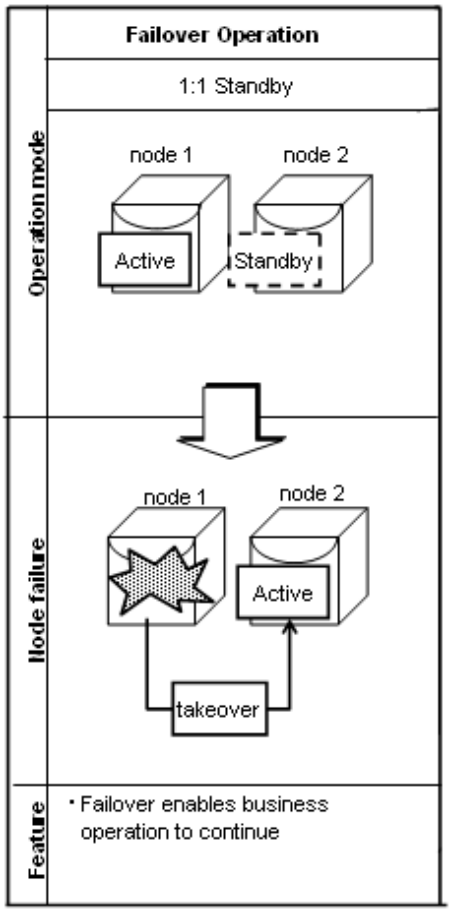
Failover operation supports the standby feature.

When an abnormality occurs on a node, the standby feature starts the instance that is the standby system at the time of the switch, and transfers the job to it.

Operation mode of failover operation

The operation mode supported in failover operation is 1:1 standby.

1:1 standby is a mode in which one active system or standby system operates on one node. Because failover allows jobs to continue when an abnormality occurs in the active system, this mode allows highly reliable systems to be constructed.



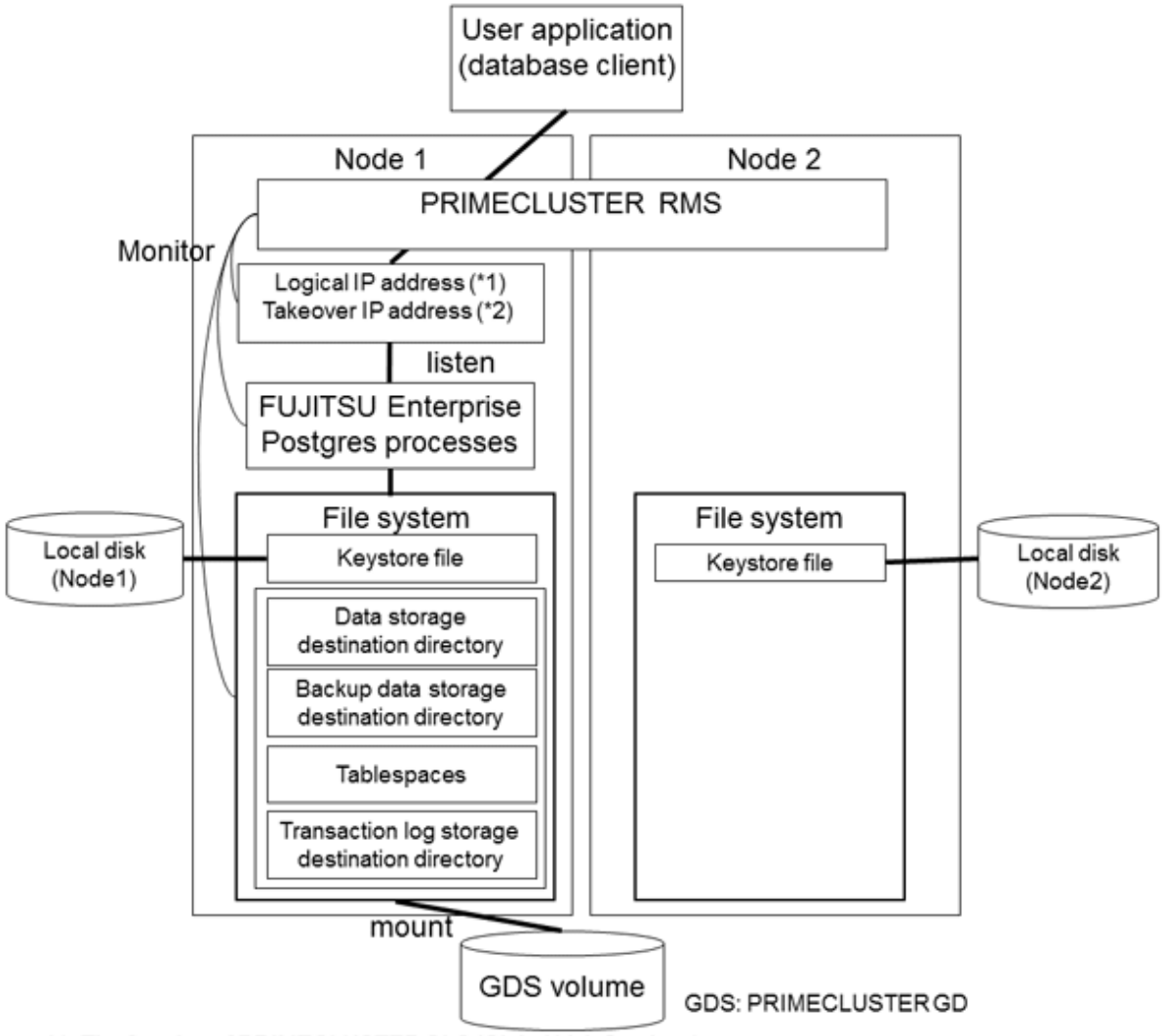
1.2 System Configuration of Failover Operation

In the illustration below, the file system is on a GDS volume (shared disk), mounted on node 1 only.

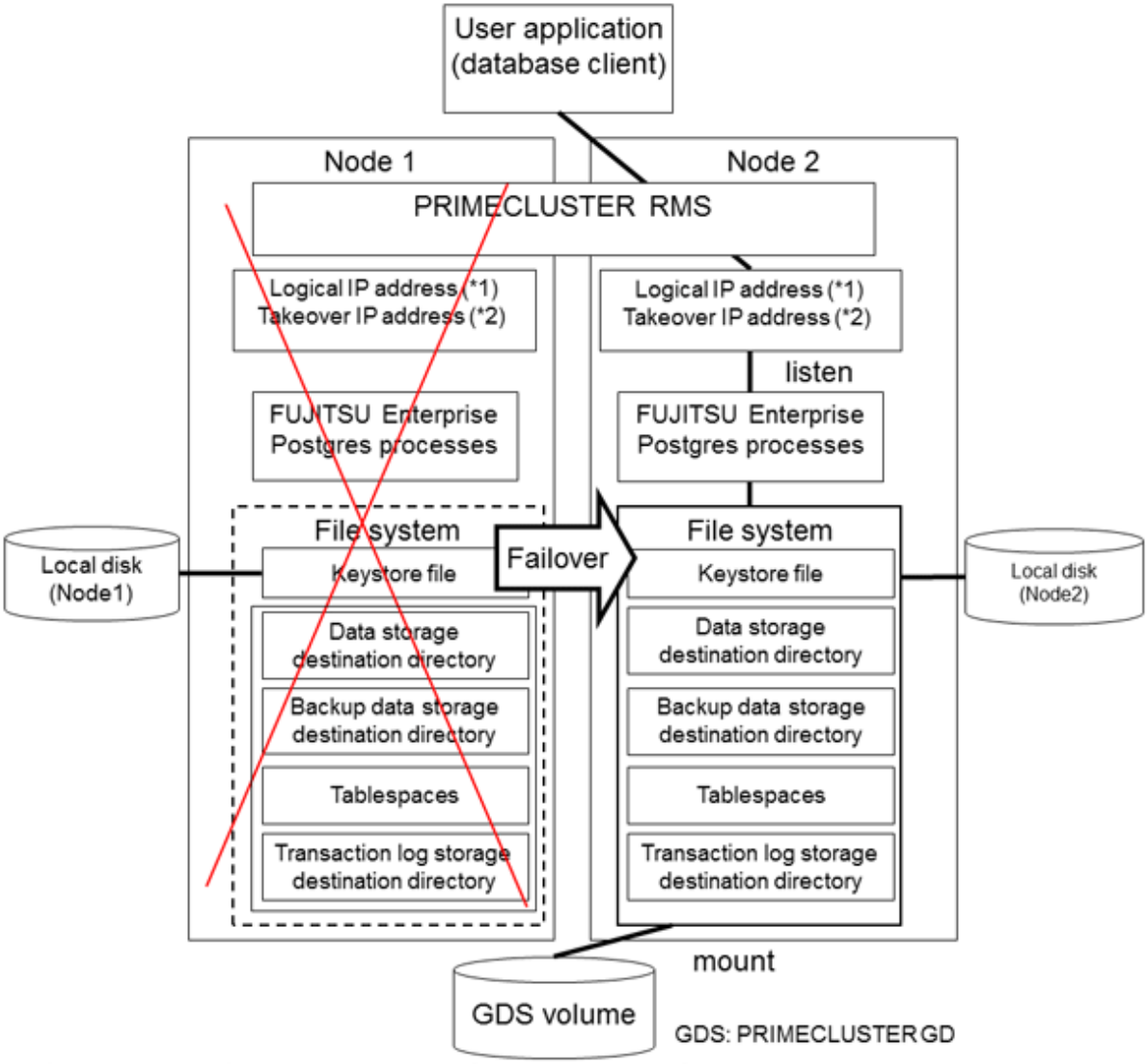
The following Fujitsu Enterprise Postgres resources are located in this file system:

- Data storage destination directory
- Tablespaces
- Backup data storage destination directory
- Transaction log storage destination directory

A Fujitsu Enterprise Postgres process waits for connections of user applications that access the database, on a logical IP address activated on node 1 only or on a takeover IP address.



When an error is detected on node 1, PRIMECLUSTER mounts the file system on node 2, and activates the logical IP address or the takeover IP address on it. The Fujitsu Enterprise Postgres process is also started on node 2. Since the logical IP address or the takeover IP address remains unchanged, the user application can access Fujitsu Enterprise Postgres process on node 2 simply by re-establishing the database connection.



*1: The function of PRIMECLUSTER GLS (Global Link Services)
*2: The function of PRIMECLUSTER Taking over network resource

Chapter 2 Setting Up Failover Operation

Use the procedure in the table below to perform setup:

Step	Work item		Reference
	Active system	Standby system	
1	Installing PRIMECLUSTER and Fujitsu Enterprise Postgres		2.1 Installing PRIMECLUSTER and Fujitsu Enterprise Postgres
2	Configuring PRIMECLUSTER		2.2 Configuring PRIMECLUSTER
3	Creating a GDS volume (*1)		2.3 Creating a GDS Volume
4	Creating a file system (*1)		2.4 Creating a File System
5	Creating a user for the operating system to start Fujitsu Enterprise Postgres		2.5 Creating an Operating System User to Start Fujitsu Enterprise Postgres
6	Mounting the file system		2.6 Mounting the File System
7	Creating a database cluster(*1)		2.7 Creating a Database Cluster
8	Registering resource information for the Fujitsu Enterprise Postgres database cluster		2.8 Registering Resource Information for the Fujitsu Enterprise Postgres Database Cluster
9	Configuring storage data protection using transparent data encryption (*1)		2.9 Configuring Storage Data Protection Using Transparent Data Encryption
10	Unmounting the file system		2.10 Unmounting the File System
11	Creating a cluster application		2.11 Creating a Cluster Application
12	Application development		2.12 Application Development
13	Checking operation		2.13 Checking Operation

*1: Some settings and operations must also be performed on the standby node. Refer to the relevant sections for details.

Note: GDS: PRIMECLUSTER GD

The sections that follow explain each step.

2.1 Installing PRIMECLUSTER and Fujitsu Enterprise Postgres

Refer to the manual for each product, and then install each product.

See

- Refer to the PRIMECLUSTER Installation and Administration Guide for information on how to install PRIMECLUSTER.
- Refer to the Installation and Setup Guide for Server for information on how to install Fujitsu Enterprise Postgres.

2.2 Configuring PRIMECLUSTER

Configure the initial settings for the cluster.

If using PRIMECLUSTER GL (hereafter referred to as "GLS"), configure the GLS settings.

See

Refer to the PRIMECLUSTER Installation and Administration Guide for information on steps required before setup.

2.3 Creating a GDS Volume

Each of the following Fujitsu Enterprise Postgres resources must be located in a file system on a shared disk:

- Data storage destination directory
- Tablespaces
- Backup data storage destination directory
- Transaction log storage destination directory

Note

- Store data storage destination directories or tablespaces and backup data storage destination directories for these in different GDS classes to guard against file system corruption and the MONITORONLY attribute settings at the time of the Gds resource and Fsystem resource creation.
- It is also recommended that database storage destination directories and transaction log storage destination directories be placed in different groups or different single disks, so that the I/O load is distributed.
- Create a directory for each resource on each node. Ensure that the directories to be mounted have the same path on all nodes.

See

- Refer to the PRIMECLUSTER Installation and Administration Guide for information on GDS volume creation.
- Refer to "Preparing Directories to Deploy Resources" in the Installation and Setup Guide for Server for information on how to create directories.

2.4 Creating a File System

Create a file system.

See

- Refer to the PRIMECLUSTER Installation and Administration Guide for information on how to create a file system on a GDS volume.

2.4.1 Creating a File System

Use the operating system or file system command to create a file system on the GDS volume.

Example)

The example below creates a file system using ext4.

```
# mkfs -t ext4 /dev/sfdsk/class0001/dsk/volume0001
```

2.4.2 Editing Configuration Files

Edit /etc/fstab and /etc/fstab.pcl on each node.

- /etc/fstab

  Comment out each line below by prepending it with "#".

```
#/dev/sfdsk/class0001/dsk/volume0001 /mnt/swdsk1 ext4 noauto 0 0
#/dev/sfdsk/class0001/dsk/volume0002 /mnt/swdsk2 ext4 noauto 0 0
```

- /etc/fstab.pcl

  Copy the entries above, and then replace "#" in each line with "#RMS#", as follows:

```
#RMS#/dev/sfdsk/class0001/dsk/volume0001 /mnt/swdsk1 ext4 noauto 0 0
#RMS#/dev/sfdsk/class0001/dsk/volume0002 /mnt/swdsk2 ext4 noauto 0 0
```

Note

Different files are specified depending on the version of the PRIMECLUSTER. Check the version of PRIMECLUSTER being used and edit appropriate files. Refer to the PRIMECLUSTER Installation and Administration Guide for details.

2.4.3 Checking

Start the GDS volume, and ensure that each file system can be mounted on each node.

```
# mount -t ext4 /dev/sfdsk/class0001/dsk/volume0001 /mnt/swdsk1
# mount -t ext4 /dev/sfdsk/class0001/dsk/volume0002 /mnt/swdsk2
# umount /mnt/swdsk1
# umount /mnt/swdsk2
```

2.5 Creating an Operating System User to Start Fujitsu Enterprise Postgres

Create an operating system user (an OS user that will become the instance administrator) who is to start Fujitsu Enterprise Postgres.

Note

Ensure that the name, UID, and GID of the operating system user (an OS user that will become the instance administrator) who is to start Fujitsu Enterprise Postgres match on all nodes that make up the cluster system.

2.6 Mounting the File System

1. Start the GDS volume if it has not already started.
2. Mount all file systems containing the resources required to create database clusters, such as for the data storage destination directory, as shown below:

```
# mount -t ext4 /dev/sfdsk/class0001/dsk/volume0001 /mnt/swdsk1
```

Note

Create directories in the file system for storing each resource. Each directory must meet certain conditions to be able to create the Fujitsu Enterprise Postgres database cluster. Check the notes in "Preparing Directories to Deploy Resources" in the Installation and Setup Guide for Server.

2.7 Creating a Database Cluster

1. Create the database cluster.

   ### Note

   - If the configuration requires password authentication, create a .pgpass file in the home directory of the operating system user who is to start Fujitsu Enterprise Postgres, and then specify a password for the template1 database as this is required. This will be used when PRIMECLUSTER checks the operation of the Fujitsu Enterprise Postgres database server. If authentication fails, a fault will be assumed to have occurred in the database server.
   - The value specified for the port parameter in postgresql.conf needs to be set in the /etc/services files on all nodes.
   - Since postgresql.conf is stored in the data directory, it becomes a common configuration file for each node in the shared disk. Therefore, it must be taken into account that the path set for postgresql.conf must be valid for the Fujitsu Enterprise Postgres server processes started on each node. For example, if a local directory is specified in the core_directory parameter, that local directory must have been created on all nodes.
   - To execute the Fujitsu Enterprise Postgres commands, you must configure the environment variables on each node.
   - Use the initdb command to create a database cluster.
   - Specify "off" for the restart_after_crash parameter in postgresql.conf.
     If "on" is specified, or the default value is used for the restart_after_crash parameter, behaviors equivalent to restarting the Fujitsu Enterprise Postgres, including crash recovery, will be performed when some server processes go down. When linking with PRIMECLUSTER, however, a failover will occur after an error is detected as soon as some server processes go down, and server processes restarting will be forced to stop. This is to inhibit complex behaviors such as processes being canceled in the middle of restarting for no particular meaningful purpose.

2. Check starting, connection, and stopping at each node. The procedure is as follows.

   ### Note

   Queries are issued to monitor the Fujitsu Enterprise Postgres from PRIMECLUSTER. With this connection process, if a password is requested, the monitoring process cannot be continued. For this reason, when checking the connection, ensure that connections can be made without entering a password.

   ### Point

   When monitoring Fujitsu Enterprise Postgres from PRIMECLUSTER, the name of the application that connects to instances is "pgx_wch_svprocess".

1. Check the starting, connection, and stopping at node 1.

   Use the pg_ctl command to start and stop the node.

   For connection, execute the following to ensure that connections can be made without entering a password:

   ```
   su - OsUser
   psql -d template1 -p portNum [ -U dbUser]
   ```

   OsUser: Operating system user to start the Fujitsu Enterprise Postgres
   portNum: Port number specified when the database cluster was created
   dbUser: Database superuser

   If the database superuser is specified at the time of executing the initdb command, or if the database superuser has been changed after the initdb command was executed, specify the user for the -U option. If the database superuser is not specified or changed, omit this.

2. Unmount all file systems containing the resources required to create database clusters (such as the data storage destination directory) on node 1, and stop the GDS volume.

3. Start the GDS volume on node 2, and mount all file systems containing the resources required to create database clusters (such as the data storage destination directory).

4. Check starting, connection, and stopping at node 2.

   Perform the same check as step 1.

5. Unmount all file systems containing the resources required to create database clusters (such as the data storage destination directory) on node 2, and stop the GDS volume.

6. Start the GDS volume on node 1, and mount all file systems containing the resources required to create database clusters (such as the data storage destination directory).

- Refer to "Using the initdb Command" of "Creating an Instance" in the Installation and Setup Guide for Server" for information on how to create a database cluster.
- Refer to "The Password File" in "Client Interfaces" in the PostgreSQL Documentation for information on files for which passwords are set.

2.8 Registering Resource Information for the Fujitsu Enterprise Postgres Database Cluster

1. Use the pgx_pclrsc command to register the database cluster in PRIMECLUSTER - the following is the simplest execution example: The database cluster does not need to be started when executing the command.

   ```
   # pgx_pclrsc -a -c databaseClusterName -u OsUser -D pgData -w workDir -n memberNodes
   ```

Item	Explanation
databaseClusterName	Specify the Fujitsu Enterprise Postgres database cluster name. The database cluster name is case-sensitive. The database cluster name must be as follows: - Up to 16 bytes, and - The first byte must be an ASCII letter, and - The other bytes must be ASCII alphanumeric characters, or underscores (_).

Item	Explanation
OsUser	Specify the operating system user (an OS user that will become the instance administrator) who can start the Fujitsu Enterprise Postgres database server.
pgData	Specify the absolute path of the data storage destination directory specified during database cluster creation.
workDir	Specify the directory for storing the temporary files required for Fujitsu Enterprise Postgres resource monitoring and state transition, and for storing the trace logs required in case of problems. If a parent directory that does not exist is specified, create the directory. The newly created directory path assumes the *OsUser* specified with the -u option as the owner, and 0700 as the privilege Required directory size is twice the size that is specified in the --trace-max-file-size option. If the default value is specified in the --trace-max-file-size option, 10240 KB x 2 is required for the directory size.
memberNodes	Specify the names of all nodes that make up the cluster system. Use a comma (,) to separate each node name. Suffix each cluster node name with the literal "RMS". Specification example: -n node1RMS,node2RMS

Other than the above, the following options can be specified:

Option	Explanation
--db-user=name	Specify the Fujitsu Enterprise Postgres database superuser. The user specified in the -u option is the default.
--response-timeout=seconds	In cluster operations, queries are regularly issued to the server to perform error detection and state monitoring. Specify a timeout time for queries issued for the heartbeat monitoring of the Fujitsu Enterprise Postgres database server. For queries, "SELECT 1" for the database "template1" is used. If 0 is specified, queries will not time out. The default is 0.
--timeout-retry-count=count	Specify the upper limit for the number of retries when queries for monitoring time out. The default is 6 times.
--trace-max-file-size	Specify the maximum size of the trace file. The default is 10240 KB.

Option	Explanation
--watch-interval=seconds	Specify the monitoring interval (the interval with which the queries are issued to the server) for a monitoring program to monitor the Fujitsu Enterprise Postgres database server. The default is 3 seconds.

See

Refer to the "B.1 pgx_pclrsc" for information on the pgx_pclrsc command.

2.9 Configuring Storage Data Protection Using Transparent Data Encryption

If using transparent data encryption, refer to "Protecting Storage Data Using Transparent Data Encryption" in the Operation Guide to configure it. Take note on the following when configuring it:

For file-based keystores

Location of the keystore file

Do not specify a shared disk for the keystore_location parameter in postgresql.conf. Instead, specify a local directory with the same path on all nodes that comprise the cluster application.

If a shared disk is specified, startup from the cluster application will fail. This is because the keystore.aks file, which is generated when automatic opening of the keystore is enabled, is obfuscated so that it can only be read on the node where it was generated, and therefore it cannot be shared across the shared disk.

Distribution of the keystore file

The master encryption key settings must be common across all nodes. For this reason, configure the master encryption key settings on one node, and then copy the keystore file to each node. Also, if the master encryption key or passphrase is changed, you must also copy the keystore file to each node.

Enabling automatic opening of the keystore

Automatic opening of the keystore must be enabled in all nodes that comprise the cluster application. Note that, when enabling the automatic opening of the keystore, only the directory in which the keystore file is stored will be referenced, therefore tasks such as starting and stopping the GDS volume are not required.

If it is not enabled, then startup from the cluster application will fail. This is because the startup process will not finish, since it waits for the manual input of the passphrase that opens the keystore.

Using a key management system

Arrangement of connection information file for key management system

The file describing the key management system connection information specified in the tde_kms.kms_conninfo_file parameter of the postgresql.conf file and the files referenced by that file, such as certificates, can be placed on a shared disk. However, the obfuscated credential file used to enable automatic keystore opening should be stored in a file on a local disk with the same path on all nodes that make up the cluster application.

Enable auto open keystore

Be sure to enable automatic keystore opening. Enable automatic keystore opening on all nodes that make up the cluster application. Note that you do not need to start or stop the GDS volume, because enabling automatic keystore opening refers only to the directory where the obfuscated credential file resides.

If you have not enabled automatic keystore opening, startup from the cluster application fails. because it waits for a manual pass phrase to be entered to open the keystore at startup, and does not terminate startup.

Changing credentials for key management systems

If the credentials for the key management system change, use the pgx_open_keystore function on the primary server to change the credentials. Repeat the procedure for enabling automatic keystore opening on the standby server.

2.10 Unmounting the File System

1. Unmount all file systems containing the resources required to create database clusters (such as the data storage destination directory).

```
# umount /mnt/swdsk1
```

2. Stop the GDS volume.

2.11 Creating a Cluster Application

Refer to "Appendix A Creating Resources and Creating/Modifying Cluster Applications".

2.12 Application Development

This section explains points to consider when creating applications during cluster operations.

2.12.1 Precautions when Developing Applications

To specify the IP address specified by the application, specify the IP address specified when creating the Gls resource or creating the takeover network resource. When state transition occurred, the operation can continue only by re-execute without changing the application.

Refer to "A.1.5 Creating Gls or Takeover Network Resources" for details.

2.12.2 Corrective Action with Application when State Transition Occurs

When accessing from an application running on another node such as an application server, the following describes the error information to be returned when an abnormality or the like occurs in the node where Fujitsu Enterprise Postgres operates, and the actions.

JDBC driver

State		Error information(*1)	Action
Node failure or Fujitsu Enterprise Postgres system failure	Failure occurs during access	57P01 08006	After the switch is complete, reestablish the connection, or re-execute the application.
	Accessed during node/system failure	08001	
Switch to the standby node	Switched during access	57P01 08006	
	Accessed during switch	08001	

*1: Return value of the getSQLState() of SQLException.

ODBC driver

State		Error information(*1)	Action
Node failure or Fujitsu Enterprise Postgres system failure	Failure occurs during access	57P01 08S01	After the switch is complete, reestablish the connection, or re-execute the application.
	Accessed during node/system failure	08001	
Switch to the standby node	Switched during access	57P01 08S01	
	Accessed during switch	08001	

*1: Return value of SQLSTATE.

.NET Data Provider

State		Error information	Action
Node failure or Fujitsu Enterprise Postgres system failure	Failure occurs during access	57P01 (*1) Empty string (*1) NullReferenceException is generated.	After the switch is complete, reestablish the connection, or re-execute the application.
	Accessed during node/system failure	Empty string (*1)	
Switch to the standby node	Switched during access	57P01 (*1) Empty string (*1) NullReferenceException is generated.	
	Accessed during switch	Empty string (*1)	

*1: This is the return value of the PostgresException attribute SqlState.

C library(libpq)

State		Error information	Action
Node failure or Fujitsu Enterprise Postgres system failure	Failure occurs during access	PGRES_FATAL_ERROR(*1) 57P01(*2) NULL(*2)	After the switch is complete, reestablish the connection, or re-execute the application.
	Accessed during node/system failure	CONNECTION_BAD(*3)	
Switch to the standby node	Switched during access	PGRES_FATAL_ERROR(*1) 57P01(*2) NULL(*2)	

State		Error information	Action
	Accessed during switch	CONNECTION_BAD(*3)	

*1: Return value of PQresultStatus().

*2: Return value of PQresultErrorField() PG_DIAG_SQLSTATE.

*3: Return value of PQstatus().

Embedded SQL in C

State		Error information(*1)	Action
Node failure or Fujitsu Enterprise Postgres system failure	Failure occurs during access	57P01 57P02 YE000 26000 40001	After the switch is complete, reestablish the connection, or re-execute the application.
	Accessed during node/system failure	08001	
Switch to the standby node	Switched during access	57P01 57P02 YE000 26000 40001	
	Accessed during switch	08001	

*1: Return value of SQLSTATE

Embedded SQL in COBOL

State		Error information(*1)	Action
Node failure or Fujitsu Enterprise Postgres system failure	Failure occurs during access	57P01 57P02 YE000 26000 40001	After the switch is complete, reestablish the connection, or re-execute the application.
	Accessed during node/system failure	08001	
Switch to the standby node	Switched during access	57P01 57P02 YE000 26000	

State		Error information(*1)	Action
		40001	
	Accessed during switch	08001	

*1: Return value of SQLSTATE.

2.13 Checking Operation

To ensure that the environment settings have been configured correctly, start, switch, and stop from Web-Based Admin View, and check the behavior.

To do a failover test, follow the procedure below.

1. Stop the Fujitsu Enterprise Postgres server processes by pg_ctl command with immediate mode.

   Example)

   ```
   # pg_ctl stop -m immediate
   ```

2. Check whether switching was done correctly.

Note

Fujitsu Enterprise Postgres uses the su(1) command to periodically monitor its resources.

Therefore, in a RHEL environment, after a cluster system is built, operating system messages relating to the su(1) command will be periodically output to /var/log/messages during cluster system operation.

Refer to the relevant manual of your operating system for information on how to control these messages.

Chapter 3 Failover Operation

This chapter explains the procedures involved in failover operation.

3.1 Adding Tablespaces

This section explains how to add tablespaces to a new file system.

This procedure is not required when you are adding tablespaces to an existing file system.

Perform the following steps when using a new shared disk class:

- Stop RMS on all nodes.
- Perform setup as described from "2.3 Creating a GDS Volume" to "2.4 Creating a File System".
- Modify the cluster application (refer to "A.2 Modifying Cluster Applications" for details).

Perform the following steps when using a new shared disk class:

- Stop RMS on all nodes.
- Perform the setup as described in "2.4 Creating a File System".

  If a new GDS volume is to be used, the GDS volume must be created in advance.
- Modify the cluster application (refer to "A.2 Modifying Cluster Applications" for details).

3.2 Modifying Database Cluster Resources

This section explains the following operations:

- Displaying database cluster resource information
- Modifying database cluster resource content
- Deleting a database cluster resource

Displaying database cluster resource information

Execute the pgx_pclrsc command as shown below (note that if -c is not specified, the command lists all registered database cluster names):

```
# pgx_pclrsc -p -c databaseClusterName
```

Modifying database cluster resource content

1. Stop RMS.
2. Execute the pgx_pclrsc command as shown below (this is the simplest example):

```
# pgx_pclrsc -d -c databaseClusterName
# pgx_pclrsc -a -c databaseClusterName -u OsUser -D pgData -w workDir -n memberNodes
```

Note

If you are modifying resource content, first delete the Fujitsu Enterprise Postgres resource from the cluster application, and then re-create it.

See

Refer to "B.1 pgx_pclrsc" for information on the pgx_pclrsc command.

Deleting a database cluster resource

1. Stop RMS.
2. Delete the Fujitsu Enterprise Postgres resource from the cluster application.
3. Execute the pgx_pclrsc command as shown below:

```
# pgx_pclrsc -d -c databaseClusterName
```

Note

The directory specified in the -w option of the pgx_pclrsc command during registration will not be deleted. Use operating system commands to delete it if necessary.

3.3 Operations under Normal Circumstances

Starting and stopping

For cluster operations, perform starting and stopping from the cluster application. If starting and stopping are performed by using the pg_ctl command or WebAdmin during failover operation, the cluster application will misjudge that services have gone down, resulting in unexpected behaviors.

Note

Issuing a query to monitor Fujitsu Enterprise Postgres from PRIMECLUSTER.

If a query is issued during startup or shutdown, the following message may be printed, but this is not a problem.

```
FATAL: Database system is in the process of starting
```

```
FATAL: Database system is shutting down
```

Switching

There are two ways to switch between the active node and the standby node, as shown below.

Under normal circumstances, switch using mutual switch. In an emergency, for example when there is no response from the active node, perform forced switch. Forced switch should only be used in emergency scenarios because the differences in Fujitsu Enterprise Postgres stop modes when the active node is offline will cause the statistics to be initialized, and the load will increase as crash recovery is performed after the switch.

- Mutual switch

  Refer to "3.5.2.4 Mutual Switch".

  Fujitsu Enterprise Postgres is stopped with the "fast" mode of the pg_ctl command.

- Forced switch

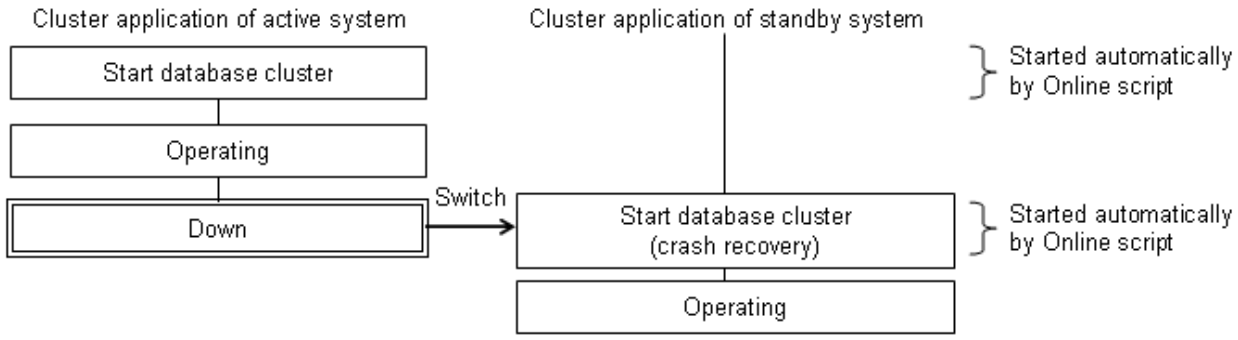
  Right-click the cluster application object, and select [Switch] from the menu. From the switchable nodes displayed in the drop-down menu, select a target node, and switch the cluster application to that node.

  Fujitsu Enterprise Postgres is stopped with the "immediate" mode of the pg_ctl command.

3.4 Operation at State Transition

The workflow for operation at state transition is shown below.

The start operation during failover operation will be performed automatically by the Online script.



Non-transferrable feature

- All transactions being processed by the cluster application on the active system will be rolled back.
- The statistics Fujitsu Enterprise Postgres has collected will be initialized.

  The statistics will be initialized in any of the following cases. Refer to the notes outlined in "Starting and Stopping an Instance" in the Operation Guide for details.

  - When the node goes down because of a kernel panic, for example
  - When forced switch is performed

    Statistics are initialized because the "immediate" mode of the pg_ctl command is used to stop Fujitsu Enterprise Postgres.

    Refer to "3.3 Operations under Normal Circumstances" for details.
  - When an error occurs in the cluster application resources

    Statistics are initialized because the "immediate" mode of the pg_ctl command is used to stop Fujitsu Enterprise Postgres to perform an immediate switch.

Errors during state transition

If an error occurs in a Fujitsu Enterprise Postgres script during cluster application state transition, the following message will be displayed in the switchlog of Cluster Admin (the registered database cluster name will be displayed in the inst1 part shown below):

```
2017-05-14 11:08:22.060:(DET, 3): ERROR: FAULT REASON: Resource <Fsep_inst1> transitioned to a
Faulted state due to a script failure.
```

To display the switchlog, select [Tools] >> [View switchlog] in the [Cluster Admin] window.

If the resource displayed in the error message is a Fujitsu Enterprise Postgres resource name, a message indicating the cause will be output either before the message above or to the system log.

3.5 Maintenance Tasks

When you need to perform tasks such as configuration changes, patch application, and hardware parts replacement that may require restarting, there are two ways of node maintenance. Performing maintenance on both nodes by stopping both active and standby nodes, and performing maintenance on one node at a time by leaving the active node running to perform maintenance on the standby node first, and then switch the active node to perform maintenance on the other node. In this section, the former is referred to as simultaneous stopped node maintenance, whereas the latter is referred to as mutual switch maintenance.

Refer to the tasks outlined below in either "Figure 3.1 Workflow for simultaneous stopped node maintenance" or "Figure 3.2 Workflow for mutual switch maintenance", and perform your maintenance tasks accordingly.

Figure 3.1 Workflow for simultaneous stopped node maintenance

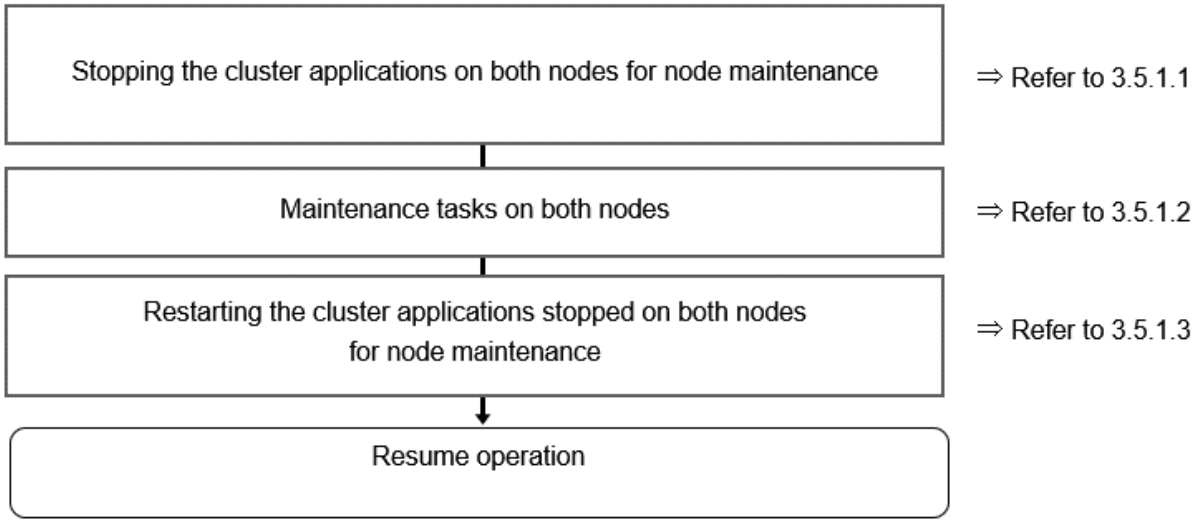
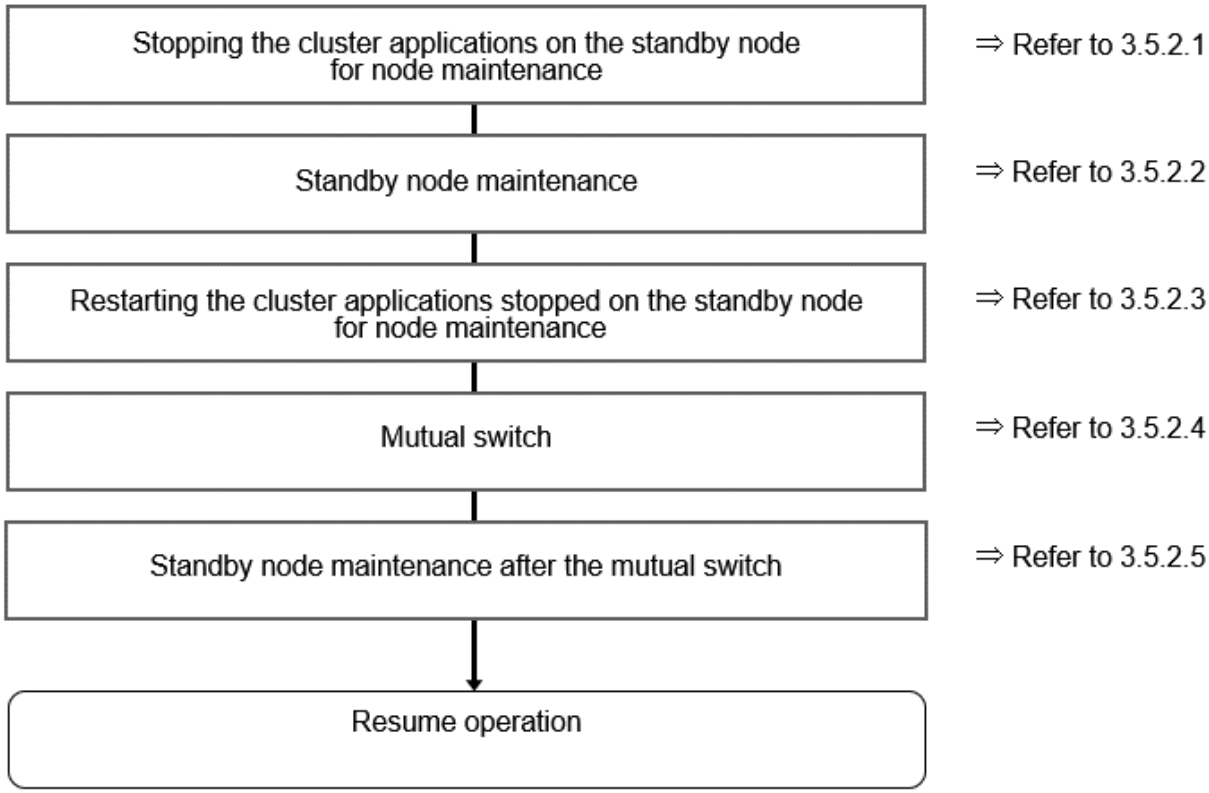


Figure 3.2 Workflow for mutual switch maintenance



3.5.1 Simultaneous Stopped Node Maintenance

This section explains the procedure to perform maintenance on both active and standby nodes simultaneously.

3.5.1.1 Stopping the Cluster Applications on Both Nodes for Node Maintenance

Follow the steps shown below to stop the cluster application on the active and standby nodes:

1. Start the [Cluster Admin] window.

2. Stop the cluster application.

   On the RMS tree in the [Cluster Admin] window, right-click the standby cluster application to be stopped, and select [Offline] from the pop-up menu to stop it.
   Stop the cluster application on both the active and standby nodes.

3. Stop RMS.

   On the RMS tree in the [Cluster Admin] window, right-click the system node where the cluster application to be stopped exists, and select [Shutdown RMS] from the pop-up menu to stop RMS. To perform maintenance without stopping RMS, execute the hvutil -m on command to change the cluster application to maintenance mode.

   Note

   - If applying the Fujitsu Enterprise Postgres update patches, RMS on the node (onto which the patches are applied) must be stopped. This is because the Fujitsu Enterprise Postgres monitoring process that runs when linked with PRIMECLUSTER must be stopped.
   - By stopping RMS, all cluster applications on the system node where RMS is stopped will stop. When you have multiple cluster applications configured, ensure that stopping all other cluster applications will not cause any problems, before stopping RMS.

3.5.1.2 Maintenance Tasks on Both Nodes

Perform maintenance tasks such as configuration changes, patch application, hardware and parts replacement that may require restarting.

3.5.1.3 Restarting the Cluster Applications Stopped on Both Nodes for Node Maintenance

Follow the steps shown below to restart the cluster application on the active and standby nodes.

1. Start the [Cluster Admin] window.
2. If RMS has not been started, select [Tools] >> [Start RMS] in the [Cluster Admin] window, and select the node to be started and click [Ok].

   Note that if the cluster applications were stopped for node maintenance using the hvutil -m on command, execute the hvutil -m off command instead of starting RMS.

3. Start the cluster application.

   On the RMS tree in the [Cluster Admin] window, right-click the standby cluster application, and from the pop-up menu, if you are starting the node as the active node, select [Online], or if you are starting it as the standby node, select [Standby]. If the state is [Fault], then select [Clear fault].

   This operation is not required if the cluster application has been configured to start automatically when RMS is started.

3.5.2 Mutual Switch Maintenance

This section explains the procedure to perform maintenance on the standby node while running the active node at the same time.

3.5.2.1 Stopping the Cluster Applications on the Standby Node for Node Maintenance

Follow the steps shown below to stop the cluster application on the standby node only:

1. Edit the /opt/SMAW/SMAWRrms/bin/hvenv.local file, and add "export HV_RCSTART=0".

   If the hvenv.local file does not exist, then create it.

   ```
   # vi /opt/SMAW/SMAWRrms/bin/hvenv.local
   Add export HV_RCSTART=0
   ```

2. On the RMS tree in the [Cluster Admin] window, right-click the cluster application on the standby node to be stopped, and select [Offline] from the pop-up menu to stop it.

3. On the RMS tree in the [Cluster Admin] window, right-click the system node (where the cluster application on the standby node that was stopped exists), and select [Shutdown RMS]. At this stage, select [Stop all Apps] as the option, and stop RMS.

Note

- Ensure that the steps up to step 3 have been completed properly before performing the maintenance tasks. If the active node goes down at the point when steps 1 and 2 have been completed, Fujitsu Enterprise Postgres will be started on the supposedly stopped standby node, and it becomes the active node instead. As a result, this will compete with the maintenance tasks and unexpected behavior may occur.
- If applying the Fujitsu Enterprise Postgres update patches, RMS on the node (onto which the patches are applied) must be stopped. This is because the Fujitsu Enterprise Postgres monitoring process that runs when linked with PRIMECLUSTER must be stopped.
- By stopping RMS, all cluster applications on the system node where RMS is stopped will stop. When you have multiple cluster applications configured, ensure that stopping all other cluster applications will not cause any problems, before stopping RMS.

3.5.2.2 Standby Node Maintenance

Perform maintenance tasks such as patch application and hardware parts replacement.

3.5.2.3 Restarting the Cluster Applications Stopped on the Standby Node for Node Maintenance

Follow the steps shown below to restart the cluster application on the standby node only (that was stopped for maintenance):

1. Edit the /opt/SMAW/SMAWRrms/bin/hvenv.local file, and delete "export HV_RCSTART=0".

```
# vi /opt/SMAW/SMAWRrms/bin/hvenv.local
Delete export HV_RCSTART=0
```

2. Select [Tools]-[Start RMS] in the [Cluster Admin] window, and select the node to be started and click [Ok].
3. On the RMS tree in the [Cluster Admin] window, right-click the cluster application to be started as the standby node, and select [Standby] from the pop-up menu to start it. If the state is [Fault], then select [Clear fault].

3.5.2.4 Mutual Switch

To perform mutual switch between the active and standby nodes, perform the steps shown below:

1. Select and right-click the cluster application on the active node, and select [Offline] from the pop-up menu to stop it.
2. Select and right-click the cluster application on the standby node, and select [Online] from the pop-up menu to start it.
3. Select and right-click the cluster application that was previously on the active node, and select [Standby] from the pop-up menu to start it as standby.

3.5.2.5 Standby Node Maintenance after the Mutual Switch

To perform maintenance tasks on the new standby node after mutual switch, perform the steps from "3.5.2.1 Stopping the Cluster Applications on the Standby Node for Node Maintenance" to "3.5.2.3 Restarting the Cluster Applications Stopped on the Standby Node for Node Maintenance".

Chapter 4 Procedures Required after a Failover Error

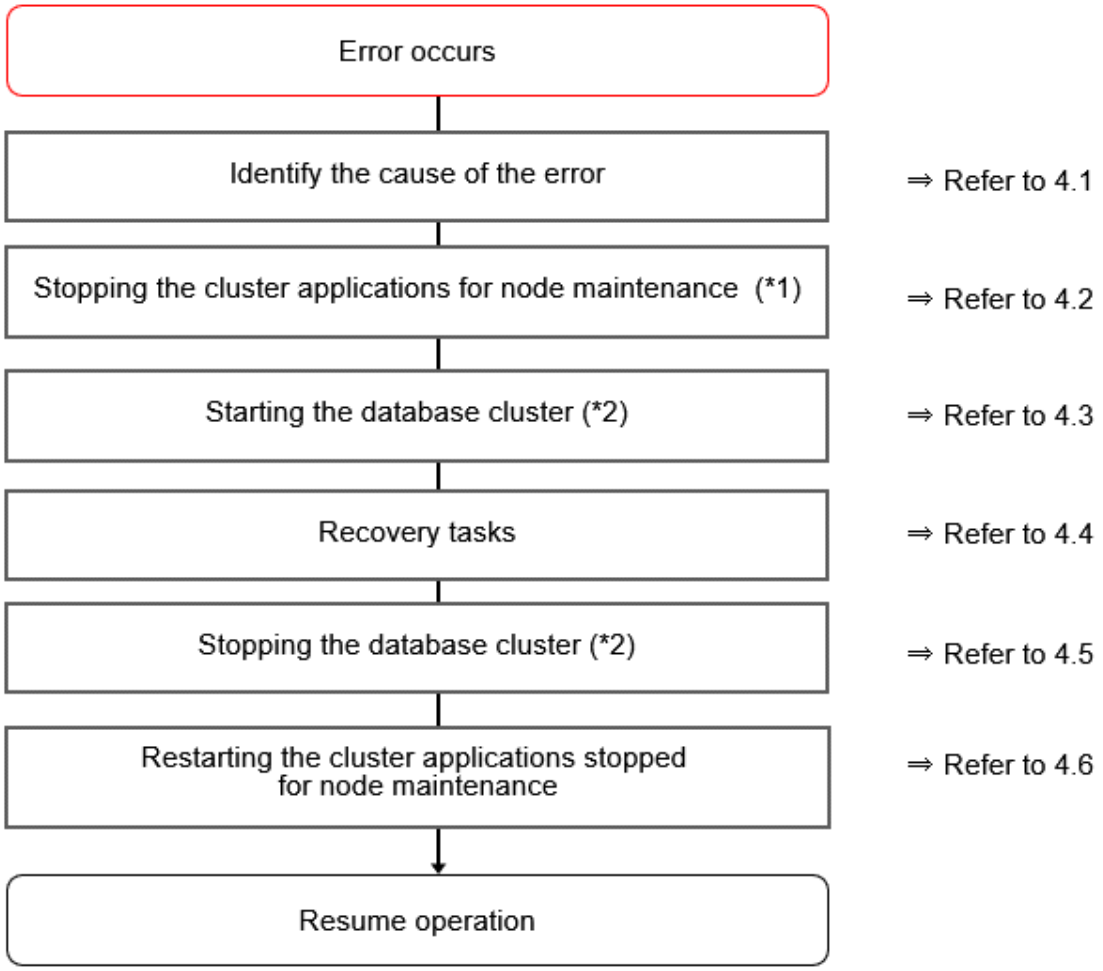
This chapter explains how to perform recovery when an abnormality occurs during failover operation.

When an error occurs during failover operation, refer to the system log to identify the cause and stop the RMS to perform recovery, separately from the RMS management. Normal operations can be resumed by restarting the RMS once the recovery is completed.

Note that the database cluster may be started or stopped without using cluster applications when performing recovery.

Refer to the tasks outlined below in "Figure 4.1 Operation flow when a failover error occurs", and perform recovery.

Figure 4.1 Operation flow when a failover error occurs



*1) If the machine is already stopped when the error occurred, this step is not necessary.

*2) Depending on the recovery tasks, start or stop the database cluster as required.

4.1 Identifying the Cause of an Error

When an error occurs during failover operation, refer to the system log to identify the cause. When cluster applications fail to start, refer to the server.log file to identify the cause.

The server.log file is a file created under the directory specified in the -w option when executing the pgx_pclrsc command. This file has almost the same content as the file specified in the -l option when executing the pg_ctl command to start the database cluster, and errors that are not output to the system log are output to these files, including startup errors.

4.2 Stopping the Cluster Applications for Node Maintenance

When an error occurs during failover operation, stop the RMS to perform recovery, separately from the RMS management. Refer to "3.5 Maintenance Tasks" for information on how to stop cluster applications for node maintenance.

Note that this task is not required in cases such as when the RMS is stopped because the machine is stopped due to an error.

4.3 Starting the Database Cluster

To perform recovery when an error occurs, it is necessary to start the database cluster without using cluster applications.

Follow the procedure below to start the database cluster if required.

Check if the database cluster needs to be started by referring to the recovery tasks written in "Actions When an Error Occurs" in the Operation Guide.

1. Start the GDS volume.
2. Mount the necessary file systems such as those where the data storage destination directory is placed.
3. Start the database cluster by using the pg_ctl command. Refer to "Using Server Commands" in "Starting an Instance and Creating a Database" in the Operation Guide for information on how to start the database cluster.

4.4 Recovery Tasks

Refer to "Actions When an Error Occurs" in the Operation Guide for information on how to perform recovery. If there are two ways available to perform recovery; using WebAdmin or commands, use commands.

4.5 Stopping the Database Cluster

If a database cluster is started, stop the database cluster.

1. Stop the database cluster using the pg_ctl command. Refer to "Using Server Commands" in "Starting an Instance and Creating a Database" in the Operation Guide" for information on how to stop the database cluster.
2. Unmount the file systems such as those where the data storage destination directory is placed.
3. Stop the GDS volume.

4.6 Restarting the Cluster Applications Stopped for Node Maintenance

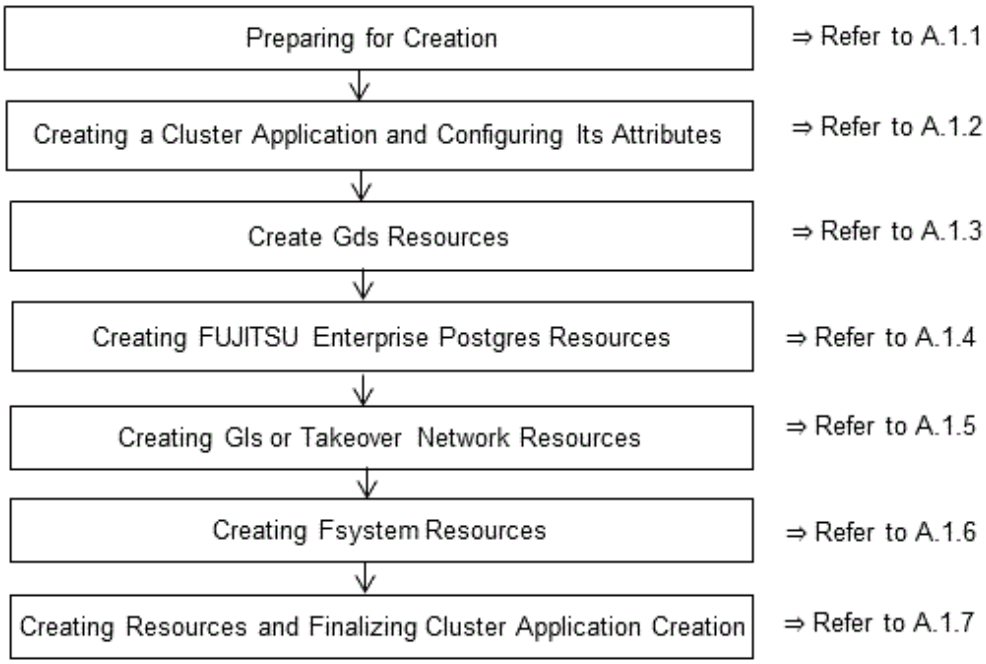
Refer to "3.5 Maintenance Tasks" to restart the cluster applications stopped for node maintenance.

Appendix A Creating Resources and Creating/Modifying Cluster Applications

This appendix explains how to create and modify cluster applications that include Fujitsu Enterprise Postgres resources in PRIMECLUSTER.

A.1 Creating Resources and Cluster Applications

Use the following procedure to create resources and a cluster application:



A.1.1 Preparing for Creation

1. Execute the hvgdsetup command for all shared disk classes to be used by all file systems to be registered in the cluster application.

```
# /opt/SMAW/SMAWRrms/bin/hvgdsetup -a class0001
```

2. From the Web-Based Admin View, stop RMS on all nodes.

3. Execute the hvw command.

```
# /opt/SMAW/SMAWRrms/bin/hvw
```

See

Refer to the PRIMECLUSTER Installation and Administration Guide for information on the hvgdsetup command.

A.1.2 Creating a Cluster Application and Configuring Its Attributes

1. Select [Application-Create].

```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
 1) HELP                                10) Configuration-Remove
```

```
 2) QUIT                                11) Configuration-Freeze
 3) Application-Create                  12) Configuration-Thaw
 4) Application-Edit                    13) Configuration-Edit-Global-Settings
 5) Application-Remove                  14) Configuration-Consistency-Report
 6) Application-Clone                   15) Configuration-ScriptExecution
 7) Configuration-Generate              16) RMS-CreateMachine
 8) Configuration-Activate              17) RMS-RemoveMachine
 9) Configuration-Copy
Choose an action: 3
```

2. Select [STANDBY].

```
Creation: Application type selection menu:
1) HELP                                 7) SCALABLE
2) QUIT                                 8) STANDBY
3) RETURN
4) OPTIONS
5) DEMO
6) GENERIC
Application Type: 8
```

3. Select [Machines+Basics].

```
Settings of turnkey wizard "STANDBY" (APP1:not yet consistent)
1) HELP                                 4) REMOVE+EXIT
2) NO-SAVE+EXIT                         5) ApplicationName=APP1
3) SAVE+EXIT                            6) Machines+Basics(-)
Choose the setting to process: 6
```

Note

Although ApplicationName can be set to any name within the range of constraints of PRIMECLUSTER, do not set it to the database cluster name specified when registering a database cluster to PRIMECLUSTER using the pgx_pclrsc command. If the same name is set, RMS will fail to start.

4. Select [AdditionalMachine].

```
Machines+Basics (app1:consistent)
 1) HELP                                14) (AutoStartUp=no)
 2) -                                   15) (AutoSwitchOver=No)
 3) SAVE+EXIT                           16) (PreserveState=no)
 4) REMOVE+EXIT                         17) (PersistentFault=0)
 5) AdditionalMachine                   18) (ShutdownPriority=)
 6) AdditionalConsole                   19) (OnlinePriority=)
 7) Machines[0]=pcl-vm13RMS             20) (StandbyTransitions=)
 8) (PreCheckScript=)                   21) (LicenseToKill=no)
 9) (PreOnlineScript=)                  22) (AutoBreak=yes)
10) (PostOnlineScript=)                 23) (AutoBreakMaintMode=no)
11) (PreOfflineScript=)                 24) (HaltFlag=no)
12) (OfflineDoneScript=)                25) (PartialCluster=0)
13) (FaultScript=)                      26) (ScriptTimeout=)
Choose the setting to process: 5
```

5. Select a standby node.

```
1) HELP
2) RETURN
3) pcl-vm13RMS
4) pcl-vm14RMS
Choose a machine for this application: 4
```

6. Ensure that all nodes that make up the cluster application are displayed in [Machines]. Note that the item displayed in Machines[0] needs to be the active node, and Machines[1] to be the standby node. If these are not set correctly, modify the setting.

```
Machines+Basics (app1:consistent)
 1) HELP                                15) (AutoStartUp=no)
 2) -                                   16) (AutoSwitchOver=No)
 3) SAVE+EXIT                           17) (PreserveState=no)
 4) REMOVE+EXIT                         18) (PersistentFault=0)
 5) AdditionalMachine                   19) (ShutdownPriority=)
 6) AdditionalConsole                   20) (OnlinePriority=)
 7) Machines[0]=pcl-vm13RMS             21) (StandbyTransitions=)
 8) Machines[1]=pcl-vm14RMS             22) (LicenseToKill=no)
 9) (PreCheckScript=)                   23) (AutoBreak=yes)
10) (PreOnlineScript=)                  24) (AutoBreakMaintMode=no)
11) (PostOnlineScript=)                 25) (HaltFlag=no)
12) (PreOfflineScript=)                 26) (PartialCluster=0)
13) (OfflineDoneScript=)                27) (ScriptTimeout=)
14) (FaultScript=)
Choose the setting to process:
```

7. Configure attributes for cluster applications from step 8.

   Refer to the PRIMECLUSTER Installation and Administration Guide for information on the attributes for cluster applications.

   Configure the attributes as shown below:

Attribute	Value
AutoStartUp (Automatically starts cluster applications when the RMS is started.)	Yes
AutoSwitchOver (Automatically switches nodes when a fault occurs in cluster applications.)	- HostFailure - ResourceFailure - ShutDown
PersistentFault (Selects whether to keep the Faulted state of the cluster application after restarting the RMS.) (*1)	0 or 1
OnlinePriority (Selects whether to use the same online nodes as those used before all clusters were restarted or follow the configured order for nodes, when all clusters were restarted.) (*2)	0 or 1
StandbyTransitions (Timing to perform standby state transition.)	- ClearFaultRequest - StartUp - SwitchRequest
HaltFlag (Selects whether to terminate the node forcibly when another error occurs during the Fault process.)	Yes

*1: If 0 is specified, the Faulted state is automatically canceled when the RMS is started. If 1 is specified, the Faulted state is kept.

*2: If 0 is specified, cluster applications will become Online in accordance with the node setting order when all clusters are restarted. If 1 is specified, cluster applications will become Online on nodes where the cluster applications were Online before the restart.

8. Select [AutoStartUp].

```
Machines+Basics (app1:consistent)
 1) HELP                              15) (AutoStartUp=no)
 2) -                                 16) (AutoSwitchOver=No)
 3) SAVE+EXIT                         17) (PreserveState=no)
 4) REMOVE+EXIT                       18) (PersistentFault=0)
 5) AdditionalMachine                 19) (ShutdownPriority=)
 6) AdditionalConsole                 20) (OnlinePriority=)
 7) Machines[0]=pcl-vm13RMS           21) (StandbyTransitions=)
 8) Machines[1]=pcl-vm14RMS           22) (LicenseToKill=no)
 9) (PreCheckScript=)                 23) (AutoBreak=yes)
10) (PreOnlineScript=)                24) (AutoBreakMaintMode=no)
11) (PostOnlineScript=)               25) (HaltFlag=no)
12) (PreOfflineScript=)               26) (PartialCluster=0)
13) (OfflineDoneScript=)              27) (ScriptTimeout=)
14) (FaultScript=)
Choose the setting to process: 15
```

9. Select [yes].

```
1) HELP
2) RETURN
3) yes
4) no
Set the AutoStartUp mode: 3
```

10. Select [AutoSwitchOver].

```
Machines+Basics (app1:consistent)
 1) HELP                              15) (AutoStartUp=yes)
 2) -                                 16) (AutoSwitchOver=No)
 3) SAVE+EXIT                         17) (PreserveState=no)
 4) REMOVE+EXIT                       18) (PersistentFault=0)
 5) AdditionalMachine                 19) (ShutdownPriority=)
 6) AdditionalConsole                 20) (OnlinePriority=)
 7) Machines[0]=pcl-vm13RMS           21) (StandbyTransitions=)
 8) Machines[1]=pcl-vm14RMS           22) (LicenseToKill=no)
 9) (PreCheckScript=)                 23) (AutoBreak=yes)
10) (PreOnlineScript=)                24) (AutoBreakMaintMode=no)
11) (PostOnlineScript=)               25) (HaltFlag=no)
12) (PreOfflineScript=)               26) (PartialCluster=0)
13) (OfflineDoneScript=)              27) (ScriptTimeout=)
14) (FaultScript=)
Choose the setting to process: 16
```

11. Select [HOSTFAILURE], then [RESOURCEFAILURE], then [SHUTDOWN], and then select [SAVE+RETURN] when the following window is displayed:

```
Set flags for AutoSwitchOver:  Currently set: HOSTFAILURE,RESOURCEFAILURE,SHUTDOWN (HRS)
1) HELP                  4) DEFAULT               7) NOT:RESOURCEFAILURE(R)
2) -                     5) NO(N)                 8) NOT:SHUTDOWN(S)
3) SAVE+RETURN           6) NOT:HOSTFAILURE(H)
Choose one of the flags: 3
```

12. Select [PersistentFault].

```
Machines+Basics (app1:consistent)
 1) HELP
 2) -
 3) SAVE+EXIT
 4) REMOVE+EXIT
 5) AdditionalMachine
 6) AdditionalConsole
```

```
 7) Machines[0]=pcl-vm13RMS
 8) Machines[1]=pcl-vm14RMS
 9) (PreCheckScript=)
10) (PreOnlineScript=)
11) (PostOnlineScript=)
12) (PreOfflineScript=)
13) (OfflineDoneScript=)
14) (FaultScript=)
15) (AutoStartUp=yes)
16) (AutoSwitchOver=HostFailure|ResourceFailure|ShutDown)
17) (PreserveState=no)
18) (PersistentFault=0)
19) (ShutdownPriority=)
20) (OnlinePriority=)
21) (StandbyTransitions=)
22) (LicenseToKill=no)
23) (AutoBreak=yes)
24) (AutoBreakMaintMode=no)
25) (HaltFlag=no)
26) (PartialCluster=0)
27) (ScriptTimeout=)
Choose the setting to process: 18
```

13. Select [0] or [1] (the example below selects [1]).

```
1) HELP
2) RETURN
3) 0
4) 1
Enable persistent fault feature for this application: 4
```

14. Select [OnlinePriority].

```
Machines+Basics (app1:consistent)
 1) HELP
 2) -
 3) SAVE+EXIT
 4) REMOVE+EXIT
 5) AdditionalMachine
 6) AdditionalConsole
 7) Machines[0]=pcl-vm13RMS
 8) Machines[1]=pcl-vm14RMS
 9) (PreCheckScript=)
10) (PreOnlineScript=)
11) (PostOnlineScript=)
12) (PreOfflineScript=)
13) (OfflineDoneScript=)
14) (FaultScript=)
15) (AutoStartUp=yes)
16) (AutoSwitchOver=HostFailure|ResourceFailure|ShutDown)
17) (PreserveState=no)
18) (PersistentFault=1)
19) (ShutdownPriority=)
20) (OnlinePriority=)
21) (StandbyTransitions=)
22) (LicenseToKill=no)
23) (AutoBreak=yes)
24) (AutoBreakMaintMode=no)
25) (HaltFlag=no)
26) (PartialCluster=0)
27) (ScriptTimeout=)
Choose the setting to process: 20
```

15. Select [0] or [1] (the example below selects [1]).

```
1) HELP
2) RETURN
3) 0
4) 1
Enable Online Priority (Active Standby) feature for this application: 4
```

16. Select [StandbyTransitions].

```
Machines+Basics (app1:consistent)
 1) HELP
 2) -
 3) SAVE+EXIT
 4) REMOVE+EXIT
 5) AdditionalMachine
 6) AdditionalConsole
 7) Machines[0]=pcl-vm13RMS
 8) Machines[1]=pcl-vm14RMS
 9) (PreCheckScript=)
10) (PreOnlineScript=)
11) (PostOnlineScript=)
12) (PreOfflineScript=)
13) (OfflineDoneScript=)
14) (FaultScript=)
15) (AutoStartUp=yes)
16) (AutoSwitchOver=HostFailure|ResourceFailure|ShutDown)
17) (PreserveState=no)
18) (PersistentFault=1)
19) (ShutdownPriority=)
20) (OnlinePriority=1)
21) (StandbyTransitions=)
22) (LicenseToKill=no)
23) (AutoBreak=yes)
24) (AutoBreakMaintMode=no)
25) (HaltFlag=no)
26) (PartialCluster=0)
27) (ScriptTimeout=)
Choose the setting to process: 21
```

17. Select [CLEARFAULTREQUEST], then [STARTUP], then [SWITCHREQUEST], and then select [SAVE+RETURN] when the following window is displayed:

```
Set flags for StandbyTransitions: Currently set: ClearFaultRequst,StartUp,SwitchRequest (CTW)
1) HELP                                5) NOT:CLEARFAULTREQUEST(C)
2) -                                   6) NOT:STARTUP(T)
3) SAVE+RETURN                         7) NOT:SWITCHREQUEST(W)
4) NO(N)
Choose one of the flags: 3
```

18. Select [HaltFlag].

```
Machines+Basics (app1:consistent)
 1) HELP
 2) -
 3) SAVE+EXIT
 4) REMOVE+EXIT
 5) AdditionalMachine
 6) AdditionalConsole
 7) Machines[0]=pcl-vm13RMS
 8) Machines[1]=pcl-vm14RMS
 9) (PreCheckScript=)
10) (PreOnlineScript=)
11) (PostOnlineScript=)
```

```
12) (PreOfflineScript=)
13) (OfflineDoneScript=)
14) (FaultScript=)
15) (AutoStartUp=yes)
16) (AutoSwitchOver=HostFailure|ResourceFailure|ShutDown)
17) (PreserveState=no)
18) (PersistentFault=1)
19) (ShutdownPriority=)
20) (OnlinePriority=1)
21) (StandbyTransitions=ClearFaultRequest|StartUp|SwitchRequest)
22) (LicenseToKill=no)
23) (AutoBreak=yes)
24) (AutoBreakMaintMode=no)
25) (HaltFlag=no)
26) (PartialCluster=0)
27) (ScriptTimeout=)
Choose the setting to process: 25
```

19. Select [yes].

```
1) HELP
2) RETURN
3) yes
4) no
Set the Halt mode: 3
```

20. Ensure that all attributes are displayed as set in the previous steps, and then select [SAVE+EXIT].

```
Machines+Basics (app1:consistent)
 1) HELP
 2) -
 3) SAVE+EXIT
 4) REMOVE+EXIT
 5) AdditionalMachine
 6) AdditionalConsole
 7) Machines[0]=pcl-vm13RMS
 8) Machines[1]=pcl-vm14RMS
 9) (PreCheckScript=)
10) (PreOnlineScript=)
11) (PostOnlineScript=)
12) (PreOfflineScript=)
13) (OfflineDoneScript=)
14) (FaultScript=)
15) (AutoStartUp=yes)
16) (AutoSwitchOver=HostFailure|ResourceFailure|ShutDown)
17) (PreserveState=no)
18) (PersistentFault=1)
19) (ShutdownPriority=)
20) (OnlinePriority=1)
21) (StandbyTransitions=ClearFaultRequest|StartUp|SwitchRequest)
22) (LicenseToKill=no)
23) (AutoBreak=yes)
24) (AutoBreakMaintMode=no)
25) (HaltFlag=yes)
26) (PartialCluster=0)
27) (ScriptTimeout=)
Choose the setting to process: 3
```

21. The following window will be displayed:

```
Settings of turnkey wizard "STANDBY" (APP1:not yet consistent)
 1) HELP                              10) Enterprise-Postgres(-)
 2) -                                 11) Symfoware(-)
 3) SAVE+EXIT                         12) Procedure:SystemState3(-)
```

```
 4) -                                  13) Procedure:SystemState2(-)
 5) ApplicationName=APP1               14) Gls:Global-Link-Services(-)
 6) Machines+Basics(app1)              15) IpAddresses(-)
 7) CommandLines(-)                    16) LocalFileSystems(-)
 8) Procedure:Application(-)           17) Gds:Global-Disk-Services(-)
 9) Procedure:BasicApplication(-)
Choose the setting to process:
```

A.1.3 Creating Gds Resources

1. Select [Gds:Global-Disk-Services].

```
Settings of turnkey wizard "STANDBY" (APP1:not yet consistent)
 1) HELP                               10) Enterprise-Postgres(-)
 2) -                                  11) Symfoware(-)
 3) SAVE+EXIT                          12) Procedure:SystemState3(-)
 4) -                                  13) Procedure:SystemState2(-)
 5) ApplicationName=APP1               14) Gls:Global-Link-Services(-)
 6) Machines+Basics(app1)              15) IpAddresses(-)
 7) CommandLines(-)                    16) LocalFileSystems(-)
 8) Procedure:Application(-)           17) Gds:Global-Disk-Services(-)
 9) Procedure:BasicApplication(-)
Choose the setting to process:17
```

2. Select [AdditionalDiskClass].

```
Volume management (Gds_APP1:not yet consistent)
1) HELP                  4) REMOVE+EXIT            7) (StandbySupport=no)
2) -                     5) AdditionalDiskClass    8) (AutoRecover=no)
3) SAVE+EXIT             6) (ClassNameFilter=)     9) (Timeout=1800)
Choose the setting to process: 5
```

3. Select a class.

```
1) HELP
2) RETURN
3) FREECHOICE
4) class0001
Choose the disk class: 4
```

4. Follow the steps above for all shared disk classes to be used by all file systems to be registered in the cluster application. When doing so, configure the MONITORONLY attribute as shown in the following table:

Purpose of file system	MONITORONLY attribute
Data storage destination directory	No
Backup data storage destination directory	Yes
Table space	No
Transaction log storage destination directory	No

To set the MONITORONLY attribute to "Yes", select "MONITORONLY".

```
Set a flag for the disk class: class0001
Currently set:
1) HELP                                5) MONITORONLY(M)
2) -
```

```
3) SAVE+RETURN
4) DEFAULT
Choose additionally one of the flags:
```

5. To configure the disk class, select [StandbySupport] and specify "no" - this means that upon failure, the disk will switch according to the purpose of the class.

```
Volume management (Gds_APP1:consistent)
 1) HELP                                6) DiskClasses[0]=class0001
 2) -                                   7) (ClassNameFilter=)
 3) SAVE+EXIT                           8) (StandbySupport=no)
 4) REMOVE+EXIT                         9) (AutoRecover=no)
 5) AdditionalDiskClass                10) (Timeout=1800)
Choose the setting to process: 8
```

6. Select [SAVE+EXIT].

```
Volume management (Gds_APP1:consistent)
 1) HELP                                6) DiskClasses[0]=class0001
 2) -                                   7) (ClassNameFilter=)
 3) SAVE+EXIT                           8) (StandbySupport=no)
 4) REMOVE+EXIT                         9) (AutoRecover=no)
 5) AdditionalDiskClass                10) (Timeout=1800)
Choose the setting to process: 3
```

A.1.4 Creating Fujitsu Enterprise Postgres Resources

1. Select [Enterprise-Postgres].

```
Settings of turnkey wizard "STANDBY" (APP1:consistent)
 1) HELP                               10) Enterprise-Postgres(-)
 2) -                                  11) Symfoware(-)
 3) SAVE+EXIT                          12) Procedure:SystemState3(-)
 4) -                                  13) Procedure:SystemState2(-)
 5) ApplicationName=APP1               14) Gls:Global-Link-Services(-)
 6) Machines+Basics(app1)              15) IpAddresses(-)
 7) CommandLines(-)                    16) LocalFileSystems(-)
 8) Procedure:Application(-)           17) Gds:Global-Disk-Services(Gds_APP1)
 9) Procedure:BasicApplication(-)
Choose the setting to process:10
```

2. Select [AdditionalResource].

```
Resource (Fsep_APP1:not yet consistent)
1) HELP                                4) REMOVE+EXIT
2) -                                   5) AdditionalResource
3) SAVE+EXIT
Choose the setting to process: 5
```

3. Select the database cluster name specified in the pgx_pclrsc command in "2.8 Registering Resource Information for the Fujitsu Enterprise Postgres Database Cluster".

```
1) HELP
2) RETURN
3) FREECHOICE
4) inst1
Choose the resource: 4
```

4. Select [SAVE+EXIT].

```
Resource (Fsep_APP1:consistent)
 1) HELP                                6) ResourceName[0]=inst1
 2) -                                   7) Flags[0]=T900
```

```
 3) SAVE+EXIT                          8) OnlineTimeout[0]=3600
 4) REMOVE+EXIT                        9) OfflineTimeout[0]=1800
 5) AdditionalResource                10) FaultScript[0]=''
Choose the setting to process: 3
```

A.1.5 Creating Gls or Takeover Network Resources

Create Gls or takeover network resources to use takeover network resources. This section provides an example for configuring Gl resources.

- Refer to the PRIMECLUSTER Installation and Administration Guide for information on creating takeover network resources.
- Refer to the PRIMECLUSTER Installation and Administration Guide for information on creating Gl resources.

1. Select [Gls:Global-Link-Services].

```
Settings of turnkey wizard "STANDBY" (APP1:consistent)
 1) HELP                               10) Enterprise-Postgres(Fsep_APP1)
 2) -                                  11) Symfoware(-)
 3) SAVE+EXIT                          12) Procedure:SystemState3(-)
 4) -                                  13) Procedure:SystemState2(-)
 5) ApplicationName=APP1               14) Gls:Global-Link-Services(-)
 6) Machines+Basics(app1)              15) IpAddresses(-)
 7) CommandLines(-)                    16) LocalFileSystems(-)
 8) Procedure:Application(-)           17) Gds:Global-Disk-Services(Gds_APP1)
 9) Procedure:BasicApplication(-)
Choose the setting to process:14
```

2. Select [AdditionalTakeoverIpaddress].

```
Gls (Gls_APP1:not yet consistent)
1) HELP                                4) REMOVE+EXIT
2) -                                   5) AdditionalTakeoverIpaddress
3) SAVE+EXIT                           6) (Timeout=60)
Choose the setting to process: 5
```

3. Select a takeover IP.

```
1) HELP
2) RETURN
3) FREECHOICE
4) takeoverip
Choose a takeover IP address for Gls: 4
```

4. Select [SAVE+RETURN].

```
Set a flag for takeover IP address: takeoverip
Currently set:
1) HELP                                5) AUTORECOVER(A)
2) -
3) SAVE+RETURN
4) DEFAULT
Choose additionally one of the flags: 3
```

5. Select [SAVE+EXIT].

```
Gls (Gls_APP1:consistent)
1) HELP                                5) AdditionalTakeoverIpaddress
2) -                                   6) TakeoverIpaddress[0]=N,takeoverip
3) SAVE+EXIT                           7) (Timeout=60)
```

```
4) REMOVE+EXIT
Choose the setting to process: 3
```

A.1.6 Creating Fsystem Resources

1. Select [LocalFileSystem].

```
Settings of turnkey wizard "STANDBY" (APP1:consistent)
 1) HELP                              10) Enterprise-Postgres(Fsep_APP1)
 2) -                                 11) Symfoware(-)
 3) SAVE+EXIT                         12) Procedure:SystemState3(-)
 4) -                                 13) Procedure:SystemState2(-)
 5) ApplicationName=APP1              14) Gls:Global-Link-Services(Gls_APP1)
 6) Machines+Basics(app1)             15) IpAddresses(-)
 7) CommandLines(-)                   16) LocalFileSystems(-)
 8) Procedure:Application(-)          17) Gds:Global-Disk-Services(Gds_APP1)
 9) Procedure:BasicApplication(-)
Choose the setting to process:16
```

2. Select [AdditionalMountPoint].

```
File systems (Lfs_APP1:not yet consistent)
1) HELP                  4) REMOVE+EXIT           7) (Timeout=180)
2) -                     5) AdditionalMountPoint
3) SAVE+EXIT             6) (Filter=)
Choose the setting to process: 5
```

3. Select the mount point of the file system.

   For all mount points assigned to the data storage destination directory, the backup data storage destination directory, the tablespace directory, and the transaction log storage destination directory, repeat Steps 3 to 5. Additionally, to register multiple file systems for monitoring, repeat Steps 3 to 5 before proceeding to configure settings for file systems used for the job.

```
1) HELP                              5) /mnt/swdsk1
2) RETURN                            6) /mnt/swdsk2
3) FREECHOICE
4) ALL
Choose a mount point: 5
```

4. Configure the MONITORONLY attribute.

Purpose of the file system	MONITORONLY attribute
Data storage destination directory	No
Backup data storage destination directory	Yes
Table space	No
Transaction log storage destination directory	No

   Select MONITORONLY to specify Yes to the MONITORONLY attribute.

```
Set flags for mount point: /mnt/swdsk1 Currently set: LOCAL,AUTORECOVER (LA)
1) HELP                  4) DEFAULT               7) SHARE(S)
2) -                     5) SYNC(Y)               8) MONITORONLY(M)
3) SAVE+RETURN           6) NOT:AUTORECOVER(A)
Choose one of the flags: 8
```

5. Select [SAVE+RETURN].

```
Set flags for mount point: /mnt/swdsk1 Currently set:
LOCAL,AUTORECOVER, MONITORONLY (LAM)
1) HELP                   4) DEFAULT               7) SHARE(S)
2) -                      5) SYNC(Y)               8) NOT:MONITORONLY(M)
3) SAVE+RETURN            6) NOT:AUTORECOVER(A)
Choose one of the flags: 3
```

6. After configuring all mount points, select [SAVE+EXIT].

```
File systems (Lfs_APP1:consistent)
 1) HELP                                  6) MountPoints[0]=LA:/mnt/monitor1
 2) -                                     7) MountPoints[1]=LA:/mnt/monitor2
 3) SAVE+EXIT                             8) (Filter=)
 4) REMOVE+EXIT                           9) (Timeout=180)
 5) AdditionalMountPoint
Choose the setting to process: 3
```

7. Select [SAVE+EXIT].

```
Settings of turnkey wizard "STANDBY" (APP1:consistent)
 1) HELP                                 10) Enterprise-Postgres(Fsep_APP1)
 2) -                                    11) Symfoware(-)
 3) SAVE+EXIT                            12) Procedure:SystemState3(-)
 4) -                                    13) Procedure:SystemState2(-)
 5) ApplicationName=APP1                 14) Gls:Global-Link-Services(Gls_APP1)
 6) Machines+Basics(app1)                15) IpAddresses(-)
 7) CommandLines(-)                      16) LocalFileSystems(Lfs_APP1)
 8) Procedure:Application(-)             17) Gds:Global-Disk-Services(Gds_APP1)
 9) Procedure:BasicApplication(-)
Choose the setting to process:3
```

See

Refer to the PRIMECLUSTER Installation and Administration Guide for information on creating Fsystem resources.

A.1.7 Creating Resources and Finalizing Cluster Application Creation

1. Select [Configuration-Generate].

```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
 1) HELP                                 10) Configuration-Remove
 2) QUIT                                 11) Configuration-Freeze
 3) Application-Create                   12) Configuration-Thaw
 4) Application-Edit                     13) Configuration-Edit-Global-Settings
 5) Application-Remove                   14) Configuration-Consistency-Report
 6) Application-Clone                    15) Configuration-ScriptExecution
 7) Configuration-Generate               16) RMS-CreateMachine
 8) Configuration-Activate               17) RMS-RemoveMachine
 9) Configuration-Copy
Choose an action: 7
```

2. Select [Configuration-Activate].

   If processing is successful, "The activation has finished successfully." will be displayed. If this message is not displayed, there is an issue with the information or configuration that was set. In such a case, review the changed information.

```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
 1) HELP                                 10) Configuration-Remove
```

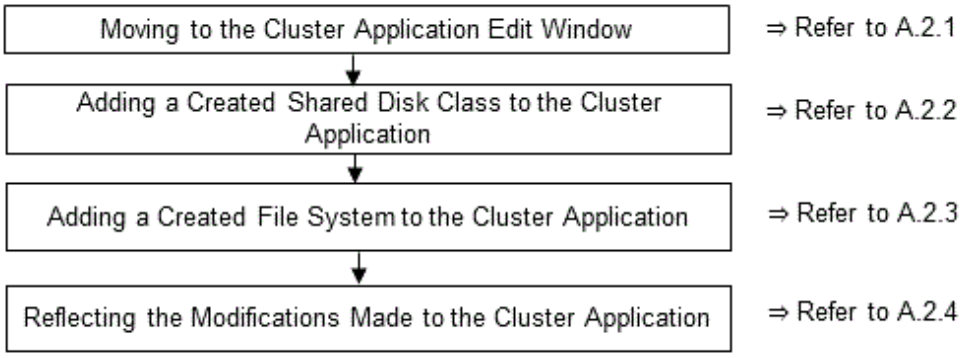
```
 2) QUIT                              11) Configuration-Freeze
 3) Application-Create                12) Configuration-Thaw
 4) Application-Edit                  13) Configuration-Edit-Global-Settings
 5) Application-Remove                14) Configuration-Consistency-Report
 6) Application-Clone                 15) Configuration-ScriptExecution
 7) Configuration-Generate            16) RMS-CreateMachine
 8) Configuration-Activate            17) RMS-RemoveMachine
 9) Configuration-Copy
Choose an action: 8
```

3. Select [QUIT].

```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
 1) HELP                              10) Configuration-Remove
 2) QUIT                              11) Configuration-Freeze
 3) Application-Create                12) Configuration-Thaw
 4) Application-Edit                  13) Configuration-Edit-Global-Settings
 5) Application-Remove                14) Configuration-Consistency-Report
 6) Application-Clone                 15) Configuration-ScriptExecution
 7) Configuration-Generate            16) RMS-CreateMachine
 8) Configuration-Activate            17) RMS-RemoveMachine
 9) Configuration-Copy
Choose an action: 2
```

A.2 Modifying Cluster Applications

Follow the steps shown below to add a new file system to a cluster application. Refer to the PRIMECLUSTER Installation and Administration Guide for information on other modifications on cluster applications. Stop the RMS on all nodes when modifying cluster applications.



A.2.1 Moving to the Cluster Application Edit Window

1. Execute the hvw command.

```
# /opt/SMAW/SMAWRrms/bin/hvw
```

2. Select [Application-Edit].

```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
 1) HELP                              10) Configuration-Remove
 2) QUIT                              11) Configuration-Freeze
 3) Application-Create                12) Configuration-Thaw
 4) Application-Edit                  13) Configuration-Edit-Global-Settings
 5) Application-Remove                14) Configuration-Consistency-Report
 6) Application-Clone                 15) Configuration-ScriptExecution
```

```
 7) Configuration-Generate              16) RMS-CreateMachine
 8) Configuration-Activate              17) RMS-RemoveMachine
 9) Configuration-Copy
Choose an action: 4
```

3. Select the user application to be used.

```
Edit: Application selection menu (restricted):
1) HELP
2) QUIT
3) RETURN
4) OPTIONS
5) APP1
Application Name: 5
```

A.2.2 Adding a Created Shared Disk Class to the Cluster Application

1. Select [Gds:Global-Disk-Services].

```
Settings of turnkey wizard "STANDBY" (APP1:consistent)
 1) HELP                                10) Enterprise-Postgres(Fsep_APP1)
 2) READONLY                            11) Symfoware(-)
 3) SAVE+EXIT                           12) Procedure:SystemState3(-)
 4) -                                   13) Procedure:SystemState2(-)
 5) ApplicationName=APP1                14) Gls:Global-Link-Services(Gls_APP1)
 6) Machines+Basics(app1)               15) IpAddresses(-)
 7) CommandLines(-)                     16) LocalFileSystems(Lfs_APP1)
 8) Procedure:Application(-)            17) Gds:Global-Disk-Services(Gds_APP1)
 9) Procedure:BasicApplication(-)
Choose the setting to process:17
```

2. Select [AdditionalDiskClass].

```
Volume management (Gds_APP1:consistent)
 1) HELP                                 6) DiskClasses[0]=class0001
 2) NO-SAVE+EXIT                         7) (ClassNameFilter=)
 3) SAVE+EXIT                            8) (StandbySupport=no)
 4) REMOVE+EXIT                          9) (AutoRecover=no)
 5) AdditionalDiskClass                 10) (Timeout=1800)
Choose the setting to process: 5
```

3. Select the created class.

```
1) HELP
2) RETURN
3) FREECHOICE
4) class0001
5) class0002
Choose the disk class: 5
```

4. Select [SAVE+EXIT].

```
Volume management (Gds_APP1:consistent)
 1) HELP                                 7) DiskClasses[1]=class0002
 2) NO-SAVE+EXIT                         8) (ClassNameFilter=)
 3) SAVE+EXIT                            9) (StandbySupport=no)
 4) REMOVE+EXIT                         10) (AutoRecover=no)
 5) AdditionalDiskClass                 11) (Timeout=1800)
 6) DiskClasses[0]=class0001
Choose the setting to process: 3
```

5. Select [SAVE+EXIT] to quit or [LocalFileSystems] to add local file systems.

```
Settings of turnkey wizard "STANDBY" (APP1:consistent)
 1) HELP                                10) Enterprise-Postgres(Fsep_APP1)
 2) -                                   11) Symfoware(-)
 3) SAVE+EXIT                           12) Procedure:SystemState3(-)
 4) -                                   13) Procedure:SystemState2(-)
 5) ApplicationName=APP1                14) Gls:Global-Link-Services(Gls_APP1)
 6) Machines+Basics(app1)               15) IpAddresses(-)
 7) CommandLines(-)                     16) LocalFileSystems(Lfs_APP1)
 8) Procedure:Application(-)            17) Gds:Global-Disk-Services(Gds_APP1)
 9) Procedure:BasicApplication(-)
Choose the setting to process:
```

A.2.3 Adding a Created File System to the Cluster Application

1. Select [LocalFileSystems].

```
Settings of turnkey wizard "STANDBY" (APP1:consistent)
 1) HELP                                10) Enterprise-Postgres(Fsep_APP1)
 2) -                                   11) Symfoware(-)
 3) SAVE+EXIT                           12) Procedure:SystemState3(-)
 4) -                                   13) Procedure:SystemState2(-)
 5) ApplicationName=APP1                14) Gls:Global-Link-Services(Gls_APP1)
 6) Machines+Basics(app1)               15) IpAddresses(-)
 7) CommandLines(-)                     16) LocalFileSystems(Lfs_APP1)
 8) Procedure:Application(-)            17) Gds:Global-Disk-Services(Gds_APP1)
 9) Procedure:BasicApplication(-)
Choose the setting to process:16
```

2. Select [AdditionalMountPoint].

```
File systems (Lfs_APP1:consistent)
 1) HELP                                 7) MountPoints[1]=LA:/mnt/swdsk2
 2) NO-SAVE+EXIT                         8) (Filter=)
 3) SAVE+EXIT                            9) (Timeout=180)
 4) REMOVE+EXIT
 5) AdditionalMountPoint
 6) MountPoints[0]=LA:/mnt/swdsk1
Choose the setting to process: 5
```

3. After registering the file systems to be used, select [SAVE+EXIT].

```
File systems (Lfs_APP1:consistent)
 1) HELP                                 8) MountPoints[2]=LA:/mnt/swdsk3
 2) NO-SAVE+EXIT                         9) (Filter=)
 3) SAVE+EXIT                           10) (Timeout=180)
 4) REMOVE+EXIT
 5) AdditionalMountPoint
 6) MountPoints[0]= LA:/mnt/swdsk1
 7) MountPoints[1]=LA:/mnt/swdsk2
Choose the setting to process:
```

A.2.4 Reflecting Modifications Made to the Cluster Application

1. Select [SAVE+EXIT].

```
Settings of turnkey wizard "STANDBY" (APP1:consistent)
 1) HELP                                10) Enterprise-Postgres(Fsep_APP1)
 2) -                                   11) Symfoware(-)
 3) SAVE+EXIT                           12) Procedure:SystemState3(-)
 4) -                                   13) Procedure:SystemState2(-)
```

```
 5) ApplicationName=APP1                14) Gls:Global-Link-Services(Gls_APP1)
 6) Machines+Basics(app1)               15) IpAddresses(-)
 7) CommandLines(-)                     16) LocalFileSystems(Lfs_APP1)
 8) Procedure:Application(-)            17) Gds:Global-Disk-Services(Gds_APP1)
 9) Procedure:BasicApplication(-)
Choose the setting to process:3
```

2. Select [RETURN].

```
Edit: Application selection menu (restricted):
1) HELP
2) QUIT
3) RETURN
4) OPTIONS
5) APP1
Application Name: 3
```

3. Select [Configuration-Generate].

```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
 1) HELP                                10) Configuration-Remove
 2) QUIT                                11) Configuration-Freeze
 3) Application-Create                  12) Configuration-Thaw
 4) Application-Edit                    13) Configuration-Edit-Global-Settings
 5) Application-Remove                  14) Configuration-Consistency-Report
 6) Application-Clone                   15) Configuration-ScriptExecution
 7) Configuration-Generate              16) RMS-CreateMachine
 8) Configuration-Activate              17) RMS-RemoveMachine
 9) Configuration-Copy
Choose an action: 7
```

4. Select [Configuration-Activate].

   If processing is successful, "The activation has finished successfully." will be displayed. If this message is not displayed, there is an issue with the information or configuration that was set. In such a case, review the changed information.

```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
 1) HELP                                10) Configuration-Remove
 2) QUIT                                11) Configuration-Freeze
 3) Application-Create                  12) Configuration-Thaw
 4) Application-Edit                    13) Configuration-Edit-Global-Settings
 5) Application-Remove                  14) Configuration-Consistency-Report
 6) Application-Clone                   15) Configuration-ScriptExecution
 7) Configuration-Generate              16) RMS-CreateMachine
 8) Configuration-Activate              17) RMS-RemoveMachine
 9) Configuration-Copy
Choose an action: 8
```

5. Select [QUIT].

```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
 1) HELP                                10) Configuration-Remove
 2) QUIT                                11) Configuration-Freeze
 3) Application-Create                  12) Configuration-Thaw
 4) Application-Edit                    13) Configuration-Edit-Global-Settings
 5) Application-Remove                  14) Configuration-Consistency-Report
 6) Application-Clone                   15) Configuration-ScriptExecution
 7) Configuration-Generate              16) RMS-CreateMachine
 8) Configuration-Activate              17) RMS-RemoveMachine
 9) Configuration-Copy
Choose an action: 2
```

Appendix B Command Reference

This appendix explains for details in command.

B.1 pgx_pclrsc

Name

pgx_pclrsc -- Register, unregister, or display Fujitsu Enterprise Postgres database cluster with PRIMECLUSTER as resource of PRIMECLUSTER.

Synopsis

```
pgx_pclrsc  -a  -c  name  -u  osuser  -D directory -w directory
            -n nodes [options...]
pgx_pclrsc -d -c name
pgx_pclrsc -p [-c name]
```

Description

pgx_pclrsc is an utility for registering or unregistering Fujitsu Enterprise Postgres database cluster with PRIMECLUSTER, or displaying settings of Fujitsu Enterprise Postgres database cluster registered with PRIMECLUSTER. Execute the command on a member node of the cluster. Only the super user(root) can execute this command.

Options

-a
--add

Register the specified database cluster with PRIMECLUSTER. If the specified database cluster has already been registered with PRIMECLUSTER, then terminate abnormally.

-c *name*
--db-cluster-name=*name*

Specify database cluster name. name is case-sensitive, and must be within 16 bytes, and an initial letter must be ascii alphabet, and following letters must be ascii alphabet or ascii digit or underscore(_).

-d
--delete

Unregister the specified database cluster from PRIMECLUSTER.

-D directory
--pgdata=*directory*

Specify an absolute path of data directory.

--db-user=*name*

Specify a database super user. Default is the user specified with '-u'.

-n nodes
--member-nodes=*nodes*

Specify names of all member nodes of the cluster. Specfity a cluster node name with uffix "RMS" to the name. Separate names with comma(,). (e.g. -n node1RMS,node2RMS)

-p
--print

Print a list of database clusters registered with PRIMECLUSTER. If -c option is specified, then print settings of the database cluster.

--response-timeout=*seconds*

Specify timeout of the query for health check. It's used with a count specified with '--timeout-retry-num'. The query is "SELECT 1" to the database "template1". If '0', wait infinitely. (default: '0')

--timeout-retry-count=*count*

Specify a limit count of retrying query when timeout is occurred. If retry counter is over the limit, then PRIMECLUSTER considers status of the database server as FAULT. If a query doesn't timeout once, retry counter is reset. (default: '6')

--trace-max-file-size=*size*

Specify max size(KB) of trace file. (default: '10240')

-u osuser
--os-user=*osuser*

Specify an OS user who can start/stop Fujitsu Enterprise Postgres database server.

-w directory
--work-dir=*directory*

Specify a directory for temporary data and trace files. It's used for starting, stopping or checking Fujitsu Enterprise Postgres database server. If the directory doesn't exist, then create it. Owner of the directory created by the command is set to the user specified with '-u'. Permission of the directory created by the command is set to 0700.

--watch-interval=*seconds*

Specify an interval. It's used for the amount of time between health checks. (default: '3')

Diagnostics

0 : On success

otherwise : On error

Notes

Before unregistering a resource, stop RMS of PRIMECLUSTER.

Example

The simplest example of registering a resource.

```
# pgx_pclrsc -a -c inst1 -u postgres -D /mnt/swdsk1/pgdata -w /var/tmp/work -n node1RMS,node2RMS
```

Appendix C Failover Operation in Streaming Replication

By combining streaming replication and failover operation, continuity of database operations can be ensured.

The following are additional notes on streaming replication when failover operation is applied to the primary server, standby server, or both.

Note

- Please refer to the relevant chapters and sections of this manual and this appendix when constructing the environment and performing operational operations.
- In this appendix, a group of nodes or a single server connected by streaming replication is referred to as a "primary server" or "standby server".
  For example, to indicate a specific node, use "the active node of the primary server" or simply "standby server" for a stand-alone server.

C.1 Possible Cluster Operation

For failover operation of primary server

Same as "1.1 Definition of Failover Operation"

For failover operation of standby server

As with the primary server, for example, if the operational system goes down, the standby system takes over the contents of the shared disk and becomes the operational system, and can continue as a standby server. Also, applications (reference operations) that you want to run on the active system of the database can be switched in conjunction with the database. Cascaded standby operation, in which replication from one standby server to another is performed, is also possible.

C.2 Overview of Failover Operation

C.2.1 Supported Operational Forms and System Configurations

The following failover operation and system configurations are supported

Operational form of streaming replication

Failover operation is available for the following servers configured for streaming replication.

- Primary server for asynchronous replication only
- Standby server for asynchronous replication
- Cascaded standby server (cascaded replication configuration)

Note

- As the reason why the synchronous replication isn't supported, if the communication path between servers is disconnected or the standby server goes down, log transfer from the primary server to the standby server will not be possible, and operations running on the primary server instance will stop. Therefore, to ensure business continuity on the primary server, synchronous replication is not supported. However, in an environment where synchronous replication is performed between servers where neither the replication source nor the replication destination is a failover operation, there is no problem in adding a server with failover operation as an additional asynchronous standby server.
- Servers that perform failover operation cannot be monitored by Mirroring Controller.
- Please refer to the "PostgreSQL Documentation" for information on how to set up and use streaming replication. This section describes the special considerations for combining streaming replication and failover operation.
- The connection status of streaming replication is not monitored by PRIMECLUSTER.

Even if streaming replication is disconnected, database operations can continue on the standby server. However, if the disconnection is prolonged, the database will not be up-to-date, so please monitor the connection status and make decisions to continue operations as necessary.

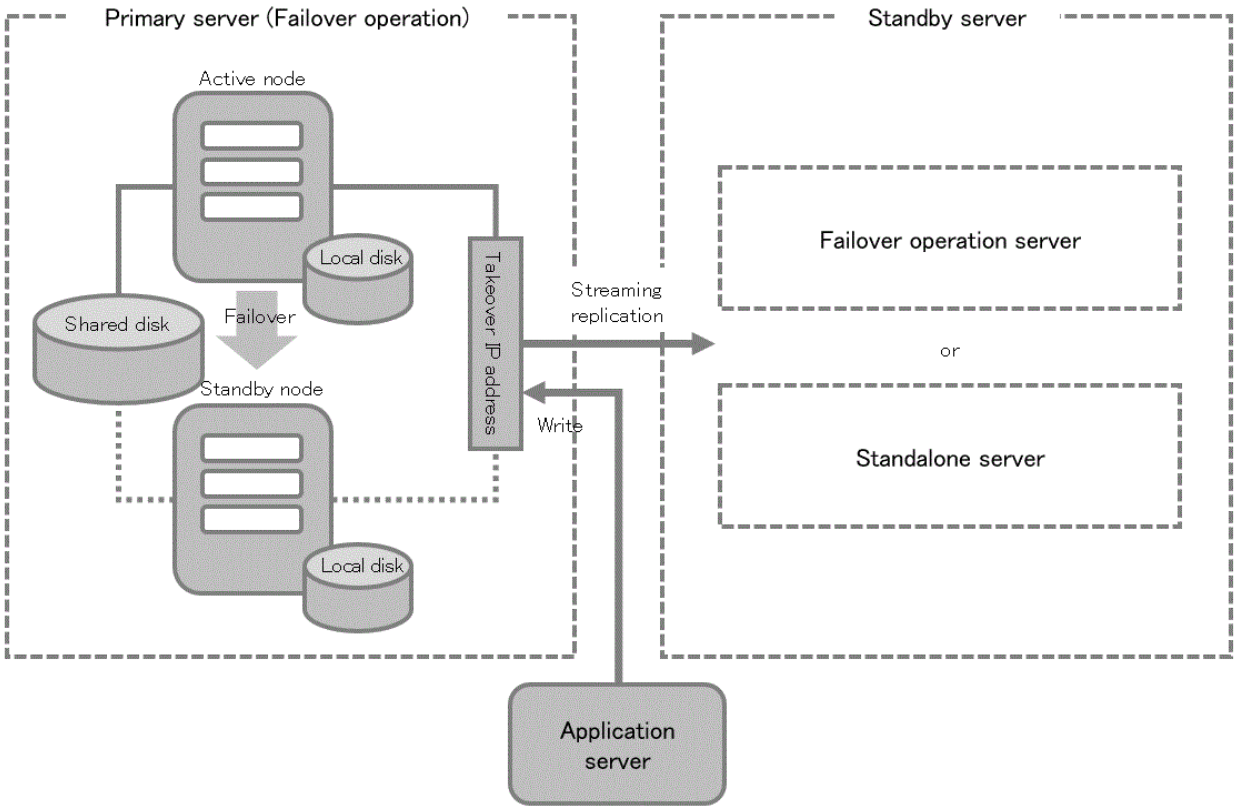
Failover operation functions and operational forms

Failover operation on the primary or standby server supports 1:1 operational standby as well as standby functionality. Cluster applications are also created on a per-primary or per-standby server basis.

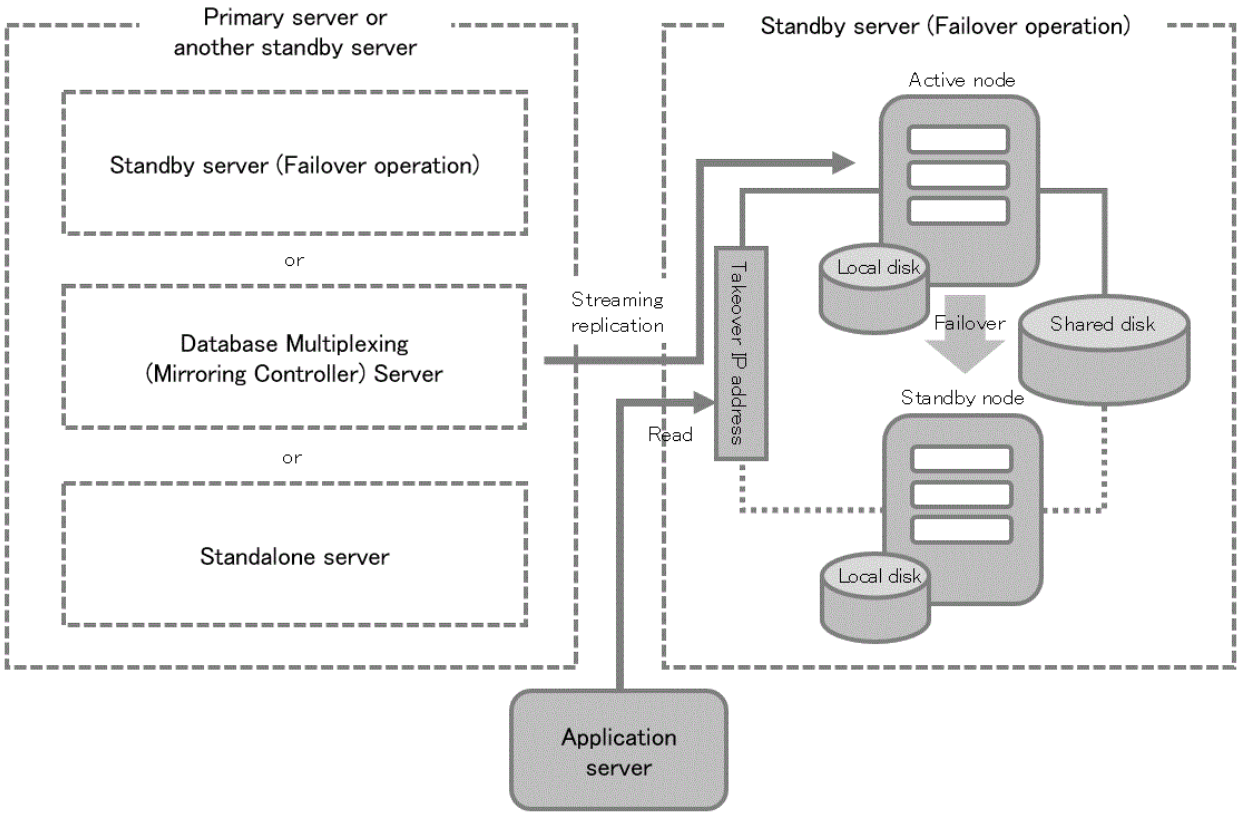
System configuration for failover operation

The following figure shows the configuration for failover operation of the primary or standby server. The primary or standby server has a file system on a shared disk that is mounted only on the active system node.

- Configuration diagram for failover operation of the primary server



- Configuration diagram for failover operation of standby server



Point

The network for connections from user applications accessing the database (business network) and the network for streaming replication connections from the standby server (log transfer network) can be located on different networks.

Enterprise Postgres resources and locations

The resources and locations of Fujitsu Enterprise Postgres on the standby and primary servers are the same as in "1.2 System Configuration of Failover Operation".

Point

The directory for storing backup data on the standby server is not required if the standby server is not promoted to the primary server.

See

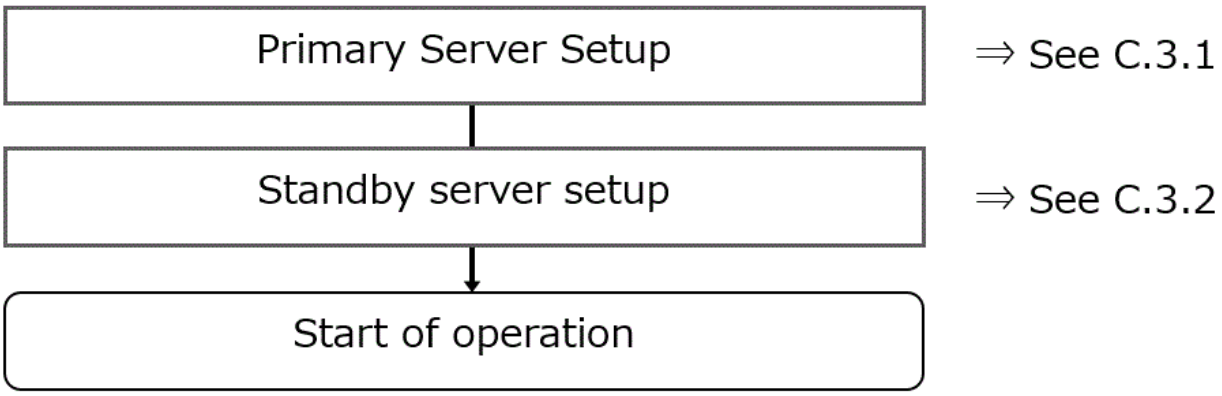
See "HA Cluster without Database Multiplexing Operation" in the Operations Guide for keystore file placement.

C.3 Set up failover operation

Follow the steps below for setup.

This procedure is described for failover operation on both the primary and standby servers.

If you want to perform the failover operation only one of the servers, please refer to the "PostgreSQL Documentation" and Fujitsu Enterprise Postgres manual for the server that will not be failovering. For the server that will perform failover operation, refer to the setup procedure for the appropriate server.



C.3.1 Primary Server Setup

Set up the primary server by referring to " Chapter 2 Setting Up Failover Operation"

However, additional precautions and steps are required in "2.2 Configuring PRIMECLUSTER", "2.7 Creating a Database Cluster" and "2.11 Creating a Cluster Application" have additional precautions and required steps.

Notes on GLS settings

Regarding the GLS settings performed in "2.2 Configuring PRIMECLUSTER", if the business network and log transfer network are separate networks, GLS settings are required for both networks.

Point

When a communication path set in the GLS is registered to a cluster, the default setting is to switch clusters in the event of a transmission path failure. While this is a desirable setting for the business network, the default setting for the log transfer network is not essential for business continuity and may result in unnecessary switching.

To avoid this problem, it is recommended that the GLS be configured not to switch clusters in the event of a transmission path failure for the log transfer network.

See

Please refer to the GLS manual for information on how to set up the cluster switching not to be performed in the event of a transmission path failure.

Note

If the setting is not to switch over in the event of a transmission path failure in the log transfer network, operations will continue with streaming replication suspended in the event of a failure in all active log transfer network. To resume streaming replication, it is necessary to identify the abnormalities and take manual measures (e.g., switching clusters, repairing failed routes, etc.).

Additional steps when creating a database cluster

The following additional steps are required for "2.7 Creating a Database Cluster".

- Configuration of pg_hba.conf file

- Configuration of postgresql.conf file

Configuration of pg_hba.conf file

Add the following entry to the pg_hba.conf file to authenticate connections from the standby server

```
# TYPE    DATABASE        USER          ADDRESS                                  METHOD
  host    replication     User name     Standby server address                  Authentication
method
```

For user name, specify the user name to connect from the standby server. Also, for Standby server address, specify the IP address to connect to the log transfer network.

The above example is for only one standby server. Please add entries for all servers you may connect as standby servers.

Point

When setting an authentication method other than trust authentication

When connecting from the standby server as the instance administrator user, create a .pgpass file in the instance administrator user's home directory to specify the password to the replication database for automatic authentication when connecting to the primary server. This will verify that the instance administrator user's OS user and the user registered in the database are the same, and that the connection was not made by an unspecified user. In addition, the pre-defined password is used for authentication to automatically connect.

Note

If trust authentication is set, all OS users who can log in to the primary server will be able to connect, and if one of these is a malicious user, then that user can corrupt the standby server data, or cause the job system to fail, by sending an erroneous transaction log. Therefore, decide on the authentication method according to the security requirements of the system using streaming replication operation.

Refer to "Authentication Methods" in the PostgreSQL Documentation for details on the authentication methods that can be set.

Configuring of postgresql.conf file

Specify the following parameters in the postgresql.conf file for streaming replication. postgresql.conf file is duplicated when the standby server instance is created. Therefore, also set the parameters required on the standby server.

Table C.1 Parameters to be set

Parameter	Contents of Designation	remarks
wal_level	replica or logical	Specify "logical" if logical decoding is also used.
max_wal_senders	Number of standby servers + 1	Specify the number of standby servers (n) + 1.
hot_standby	on	Specify this to perform reference-based operations on the standby server.
wal_keep_size	WAL storage size (megabytes)	If a delay exceeding the value set in this parameter occurs, the WAL segment required later by the primary server may be deleted. Additionally, if you stop a standby server (for maintenance, for example), consider the stop time and set a value that will not cause the WAL segment to be deleted.

Parameter	Contents of Designation	remarks
		Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on estimating the WAL save size.
wal_sender_timeout	Timeout time (milliseconds)	Specify the time to determine that an error has occurred in the transfer of transaction logs on the primary server side. By specifying this parameter, log transfer to the standby server can be resumed even if the standby server temporarily stops due to failover, etc.
wal_receiver_timeout	Timeout time (milliseconds)	Specify the time to determine that an error has occurred in receiving transaction logs on the standby server side. By specifying this parameter, log reception from the primary server can be resumed even if the primary server temporarily stops due to failover, etc.

Point

Replication slots can be used instead of the wal_keep_size parameter to prevent deletion of old WAL segments. See "Replication Slots" in the "PostgreSQL Documentation" for details on replication slots.

See

- For more information on the pg_hba.conf file, see "Client Authentication" in "PostgreSQL Documentation".
- For more information on the postgresql.conf file, see "Server Configuration" in the "PostgreSQL Documentation".

Additional steps when creating a cluster application

The following additional steps are required for "2.11 Creating a Cluster Application".

If the business network that accepts connections from applications and the log transfer network that accepts replication connections for log transferring are Gls resources on separate networks, both the business network and the log transfer network Gls resources must be Register them as cluster application resources.

C.3.2 Standby Server Setup

Follow the steps below for setup.

The procedure is almost the same as in "Chapter 2 Setting Up Failover Operation" but some of the steps are different.

In this section, the word primary server is used as the replication source server. If you are building a cascaded standby environment, please read primary server as upstream server.

Procedure	Work item		See
	Active system	standby system	
1	Installing PRIMECLUSTER and Fujitsu Enterprise Postgres		"2.1 Installing PRIMECLUSTER and Fujitsu Enterprise Postgres"
2	Configuring PRIMECLUSTER		"2.2 Configuring PRIMECLUSTER".
3	Creating a GDS Volume (*1)		"2.3 Creating a GDS Volume" Note: It is not mandatory to create a directory for storing backup data.
4	Creating a file system (*1)		"2.4 Creating a File System"
5	Creating a user for the operating system to start Fujitsu Enterprise Postgres		"2.5 Creating an Operating System User to Start Fujitsu Enterprise Postgres"
6	Mounting the file system		"2.6 Mounting the File System"
7	Creating a Database cluster (*1)		"Creating a Database Cluster"
8	Registering resource information for the Fujitsu Enterprise Postgres database cluster		"2.8 Registering Resource Information for the Fujitsu Enterprise Postgres Database Cluster".
9	Configuring storage data protection using transparent data encryption (*1)		"Creating a Database Cluster"
10	Unmounting the file system		"2.10 Unmounting the File System"
11	Creating a cluster application		"2.11 Creating a Cluster Application"
12	Application development		"2.12 Application Development"
13	Checking operation		"Operation check"

*1: Some settings and operations must also be performed on the standby node. Please refer to the reference for details.

Creating a Database Cluster

Refer to "2.7 Creating a Database Cluster" for a setup that takes into account the following

1. Add configuration for standby server to primary server

   Add the configuration items for the standby server to the database cluster configuration file created on the primary server.

   Configure the pg_hba.conf file on the primary server

   Add the following entry to the pg_hba.conf file to authenticate connections from the standby server

```
# TYPE    DATABASE        USER          ADDRESS                                 METHOD
  host    replication     username Active node address of the standby server
authentication method
  host    replication     username Standby node address of the standby server
authentication method
```

User name is the name of the user connecting from the standby server.

When stating the address of the standby server, set it to allow connections from the IP address of the respective node, rather than from the takeover IP address.

Point

When setting an authentication method other than trust authentication

When connecting from the standby server as the instance administrator user, create a .pgpass file in the instance administrator user's home directory to specify the password to the replication database for automatic authentication when connecting to the primary server. This will verify that the instance administrator user's OS user and the user registered in the database are the same, and that the connection was not made by an unspecified user. In addition, the pre-defined password is used for authentication to automatically connect.

Note

If trust authentication is set, all OS users who can log in to the primary server will be able to connect, and if one of these is a malicious user, then that

user can corrupt the standby server data, or cause the job system to fail, by sending an erroneous transaction log. Therefore, decide on the authentication method according to the security requirements of the system using streaming replication operation.

Refer to "Authentication Methods" in the PostgreSQL Documentation for details on the authentication methods that can be set.

Configure the postgresql.conf file on the primary server

Specify the following parameters in the postgresql.conf file for streaming replication. postgresql.conf file is replicated when the standby server instance is created. Therefore, set the necessary parameters in the primary server's postgresql.conf file before replicating it to the standby server. No additional configuration is required if you have already set up the same information when setting up the primary server.

Table C.2 Parameters to be set

Parameter	Contents of Designation	remarks
wal_level	replica or logical	Specify "logical" if logical decoding is also used.
max_wal_senders	Number of standby servers + 1	Specify the number of standby servers (n) + 1.
hot_standby	on	This setting must be on regardless of whether or not reference work is performed on the standby server, because the monitoring process by PRIMECLUSTER will connect to the standby server. If this setting is off, PRIMECLUSTER will erroneously assume that PostgreSQL has stopped.
wal_keep_size	WAL storage size (megabytes)	If a delay exceeding the value set in this parameter occurs, the WAL segment required later by the primary server may be deleted. Additionally, if you stop a standby server (for

Parameter	Contents of Designation	remarks
		maintenance, for example), consider the stop time and set a value that will not cause the WAL segment to be deleted. Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on estimating the WAL save size.
wal_sender_timeout	Timeout time (milliseconds)	Specify the time to determine that an error has occurred in the transfer of transaction logs on the primary server side. By specifying this parameter, log transfer to the standby server can be resumed even if the standby server temporarily stops due to failover, etc.
wal_receiver_timeout	Timeout time (milliseconds)	Specify the time to determine that an error has occurred in receiving transaction logs on the standby server side. By specifying this parameter, log reception from the primary server can be resumed even if the primary server temporarily stops due to failover, etc.

Point

Replication slots can be used instead of the wal_keep_size parameter to prevent deletion of old WAL segments. See "Replication Slots" in the "PostgreSQL Documentation" for details on replication slots.

2. Creating a Database Cluster

   To create a database cluster on the standby server, run the pg_basebackup command to create a duplicate instance of the primary server.

   Example)

```
$ pg_basebackup -D <data storage destination directory of standby server> -X fetch --
waldir=<transaction log directory> --progress --verbose -R --
dbname='application_name=<application name>' -h <takeover IP address of primary server> -p <port
number of primary server>
```

Note

- Use the pg_basebackup command with the -R option to create a standby.signal file. If you do not create the standby.signal file, the server cannot start as a standby server.
- If using a method that requires password authentication for connections to the primary server, you will need to ensure that authentication is performed automatically. If the -R option is specified for the pg_basebackup command and the password parameter is specified for the --dbname option, the pg_basebackup command will set the password in the primary_conninfo parameter in postgresql.auto.conf file, enabling connections to be performed automatically.

If a password is not set in the primary_conninfo parameter in postgresql.auto.conf file, it will be necessary to create a .pgpass file in the home directory of the instance administrator user, and specify a password for the replication database.

- The primary_conninfo parameter should not be set in the postgresql.conf file, but only in the postgresql.auto.conf file using the pg_basebackup command.
- When executing the pg_basebackup command, consider the following for collection of transaction logs.
  - When "fetch" is specified for the -X option of the command
    Transaction logs are collected at the end of the backup, so it is necessary to ensure that transaction logs that occur during backup are not deleted from the primary server. Therefore, allow for a sufficient value for the wal_keep_size parameter in postgresql.conf.

See

- For more information on the pg_basebackup command, see "pg_basebackup" in the "Reference" section of the PostgreSQL Documentation.
- For more information on the standby.signal file, see "Hot Standby" in the PostgreSQL Documentation.

3. Setting up transparent data encryption to protect stored data

   Refer to "2.9 Configuring Storage Data Protection Using Transparent Data Encryption" and consider the following in your setup

   - Keystore file location
     Specify a local directory with the same path on the active and standby nodes.
   - Keystore File Distribution
     Copy the keystore files on the active node of the primary server to both active node and standby node of the standby server.
   - Enable auto-open keystore
     Please enable in both active node and standby node of the standby server.

4. Check each node for startup, connection, and shutdown.

Operation check

Refer to "2.13 Checking Operation" Before doing so, however, make sure that streaming replication is set up correctly.

Follow the steps below.

1. On the primary server, verify that the connection status from the standby server can be retrieved by the statistics view pg_stat_replication.

   Example)

   The following is an example of output when the psql command is used.

```
postgres=# select * from pg_stat_replication;
-[ RECORD 1 ]----+------------------------------
pid              | 321621
usesysid         | 10
usename          | fsep
application_name | standby
client_addr      | 192.0.2.210
client_hostname  |
client_port      | 27500
backend_start    | 2022-12-01 19:12:53.145639+09
backend_xmin     |
state            | streaming
sent_lsn         | 0/3000060
write_lsn        | 0/3000060
flush_lsn        | 0/3000060
replay_lsn       | 0/3000060
write_lag        |
```

```
flush_lag        |
replay_lag       |
sync_priority    | 0
sync_state       | async
reply_time       | 2022-12-01 19:13:11.194744+09
```

2. Check the search results in step 1.

   Verify that a connection in asynchronous mode has been established with the intended standby server.

   Table C.3 Items to be checked

item	Confirmation details
application_name	The application name must indicate the standby server.
client_addr	It must be the IP address of the active node of the standby server.
state	streaming
sync_state	async

Note

Streaming replication status is not monitored by PRIMECLUSTER. If you need to monitor streaming replication, please create and run a monitoring program referring to this section.

C.4 Failover Operation

This section describes operations that require special attention related to streaming replication operations when performing "Chapter 3 Failover Operation".

operation		special remarks
3.1 Adding Tablespaces		If you are adding a new file system in an environment where both the primary and standby servers are in failover operation, you must add the required GDS volumes or file systems on both the primary and standby servers before executing the CREATE TABLESPACE statement.
3.2 Modifying Database Cluster Resources		-
3.3 Operations under Normal Circumstances		-
3.4 Operation at State Transition		If the active system of the primary server in failover operation goes down, the standby system will automatically start up as the primary server. When promoting the standby server to the primary server, make sure that both nodes of primary server are down before promoting it. Otherwise, the primary server and the standby server may both be ready to execute update statements.
3.5 Maintenance Tasks	3.5.1 Simultaneous Stopped Node Maintenance	When performing the simultaneous stopped node maintenance on the standby server and continuing operations on the primary server, please estimate the WAL segment taking into account the maintenance outage time.
	3.5.2 Mutual Switch Maintenance	There are no precautions in the mutual switch maintenance procedure. However, when starting maintenance work, start with the standby server.

C.5 Procedures Required after a Failover Error

C.5.1 In Case of Primary Server Error

Refer to "Chapter 4 Procedures Required after a Failover Error" for information on recovering as the primary server when the primary server has an error.

However, if it is necessary to rebuild the standby server as well, such as by restoring an instance of the primary server from backup data, refer to "C.5.2 In Case of an Error on the Standby Server" to rebuild the standby server.

After the standby server has been promoted to the new primary server, prepare for recovery by referring to "Chapter 4 Procedures Required after a Failover Error" when the former primary server that became abnormal is incorporated as a standby server and restored. However, instead of the steps from "4.3 Starting the Database Cluster" to "4.6 Restarting the Cluster Applications Stopped for Node Maintenance" please follow the steps below.

The following two commands can be used to recover standby servers. Select the one that best suits your situation and the content of the restoration.

- pg_basebackup

  Creates a copy of all resources of the primary server instance.

  Refer to "C.5.2 In Case of an Error on the Standby Server" for the rebuilding procedure.

- pg_rewind

  Creates a copy of only the updated files on the new primary server. For this reason, if this command is used to incorporate a new standby server, recovery time can be shortened. To use this command to build the original primary server as a new standby server, at least one of the following must be met:

  a. Checksums were enabled when an instance was created, or

  b. The wal_log_hints parameter of postgresql.conf was enabled when an instance was started.

  Additionally, full_page_writes must be enabled, which is its default value.

  The following is an example of running the pg_rewind command to synchronize and recover data from the old primary server with the new primary server.

  1. Start an instance of the old primary server
     Use the pg_ctl command to start the old primary server instance.

     Example)

     ```
     $ pg_ctl start -D <data storage destination directory>
     ```

  2. Stopping an instance of the old primary server
     Stop the old primary server instance using the pg_ctl command.

     Example)

     ```
     $ pg_ctl stop -D <data storage destination directory>
     ```

  3. Wait for unapplied update transaction log to be applied on new primary server
     Execute the following SQL on the new primary server and wait until the result is false.

     ```
     select pg_is_in_recovery();
     ```

     Example)

     ```
     $ psql -h <hostname of new primary server> -p <port number of new primary server> -d <database
     name> -c "select pg_is_in_recovery();"
     ```

     Any database can be connected.

Note

If pg_rewind is executed immediately after the new primary server is promoted, steps 3 and 4 are required. If the new primary server is able to execute the SQL to write and the checkpoint process is executed after the promotion, steps 3 and 4 are not necessary.

4. Update Timeline ID
   Execute a checkpoint process to update the timeline ID.

```
$ psql -h <hostname of new primary server> -p <port number of new primary server> -d <database name> -c "checkpoint;"
```

   Any database can be connected.

5. Create a duplicate instance of the new primary server on the old primary server (new standby server)

   Run the pg_rewind command to create the data on the new standby server in sync with the new primary server.

   Example)

```
$ pg_rewind -D <data storage destination directory of old primary server> -R --source-server='user=<username> host=<hostname of new primary server> port=<port number of new primary server> dbname=<database name>'
```

Note

- Use the pg_rewind command with the -R option to create a standby.signal file. If you do not create the standby.signal file, the server cannot start as a standby server.
- If using a method that requires password authentication for connections to the primary server, you will need to ensure that authentication is performed automatically. If the -R option is specified for the pg_rewind command and the password parameter is specified for the --dbname option, the pg_rewind command will set the password in the primary_conninfo parameter in postgresql.auto.conf file, enabling connections to be performed automatically.
  If a password is not set in the primary_conninfo parameter in postgresql.auto.conf file, it will be necessary to create a .pgpass file in the home directory of the instance administrator user, and specify a password for the replication database.
- If you need to set a connection string other than host, port and application_name, include it in the setting of the primary_conninfo parameter.

See

- For more information on the standby.signal file, see "Hot Standby" in the PostgreSQL Documentation.
- For more information on primary_conninfo, see "Setting Up a Standby Server" in the PostgreSQL Documentation. Setting Up a Standby Server" in the "PostgreSQL Documentation".

6. Set parameters in postgresql.conf file on old primary server (new standby server)
   Set the necessary parameters for the new standby server in the postgresql.conf file.
   The parameters to be set in the postgresql.conf file should be set with reference to "Creating a Database Cluster".

Note

Since the new primary server is branched to a new timeline by promotion, the 'latest' must be specified in the recovery_target_timeline parameter for the old primary server (new standby server) to follow the new primary server.

7. Resource Release
   Unmount the mounted file system, including the file system where the directory for data storage is located. Then, stop the GDS volume.

8. Cancellation of maintenance suspension
   Refer to "3.5 Maintenance Tasks" to cancel the maintenance suspension.

9. Operation check
   Refer to "Operation check" to check the operation.

See

- For more information on the pg_basebackup command, see "pg_basebackup" in the "Reference" section of the PostgreSQL Documentation.
- For more information on the pg_rewind command, see "pg_rewind in the "Reference" section of the "PostgreSQL Documentation".

C.5.2 In Case of an Error on the Standby Server

The following is a description of what to do in the event of a standby server malfunction.

1. Identification of the cause of the abnormality
   Identify the cause of the abnormality by referring to "4.1 Identifying the Cause of an Error"
   In case of an anomaly that requires rebuilding the standby server, refer to the following procedures. In other cases, refer to "Chapter 4 Procedures Required after a Failover Error".

2. Stop RMS on standby server, remove cluster applications and resources
   Stop RMS and then remove cluster applications and resources.

3. Rebuild standby server
   Refer to "C.3.2 Standby Server Setup" to perform the re-setup.

Index

[A]

Adding a Created File System to the Cluster Application.....38
Adding a Created Shared Disk Class to the Cluster Application37
Adding Tablespaces.....16

[C]

Checking Operation.....15
Command Reference..... 40
Configuring PRIMECLUSTER.....6
Configuring Storage Data Protection Using Transparent Data Encryption..... 11
Creating a Cluster Application..... 12
Creating a Cluster Application and Configuring Its Attributes24
Creating a Database Cluster..... 8
Creating a File System.....6
Creating a GDS Volume.....6
Creating an Operating System User to Start Fujitsu Enterprise Postgres.....7
Creating Fsystem Resources.....34
Creating Fujitsu Enterprise Postgres Resources.....32
Creating Gds Resources.....31
Creating Gls or Takeover Network Resources.....33
Creating Resources and Cluster Applications..... 24
Creating Resources and Creating/Modifying Cluster Applications.....24
Creating Resources and Finalizing Cluster Application Creation 35

[D]

Definition of Failover Operation..... 1
Deleting a database cluster resource.....17
Displaying database cluster resource information.....16

[E]

Editing Configuration Files..... 7
Errors during state transition..... 18

[F]

Failover Operation.....16
Feature of failover operation..... 2

[I]

Identifying the Cause of an Error..... 22
Installing PRIMECLUSTER and Fujitsu Enterprise Postgres... 6

[M]

Maintenance Tasks..... 18
Modifying Cluster Applications.....36
Modifying database cluster resource content..... 16
Modifying Database Cluster Resources..... 16
Mounting the File System.....8
Moving to the Cluster Application Edit Window.....36
Mutual Switch Maintenance.....20

[N]

Non-transferrable feature.....18

[O]

Operation at State Transition.....17
Operation mode of failover operation..... 2
Operations under Normal Circumstances.....17
Overview of Failover Operation.....1

[P]

Preparing for Creation..... 24
Procedures Required after a Failover Error.....22

[R]

Recovery Tasks.....23
Reflecting Modifications Made to the Cluster Application.....38
Registering Resource Information for the Fujitsu Enterprise Postgres Database Cluster..... 9
Restarting the Cluster Applications Stopped for Node Maintenance.....23

[S]

Setting Up Failover Operation.....5
Simultaneous Stopped Node Maintenance..... 19
Starting the Database Cluster..... 23
Stopping the Cluster Applications for Node Maintenance.....23
Stopping the Database Cluster.....23
System Configuration of Failover Operation..... 2

[U]

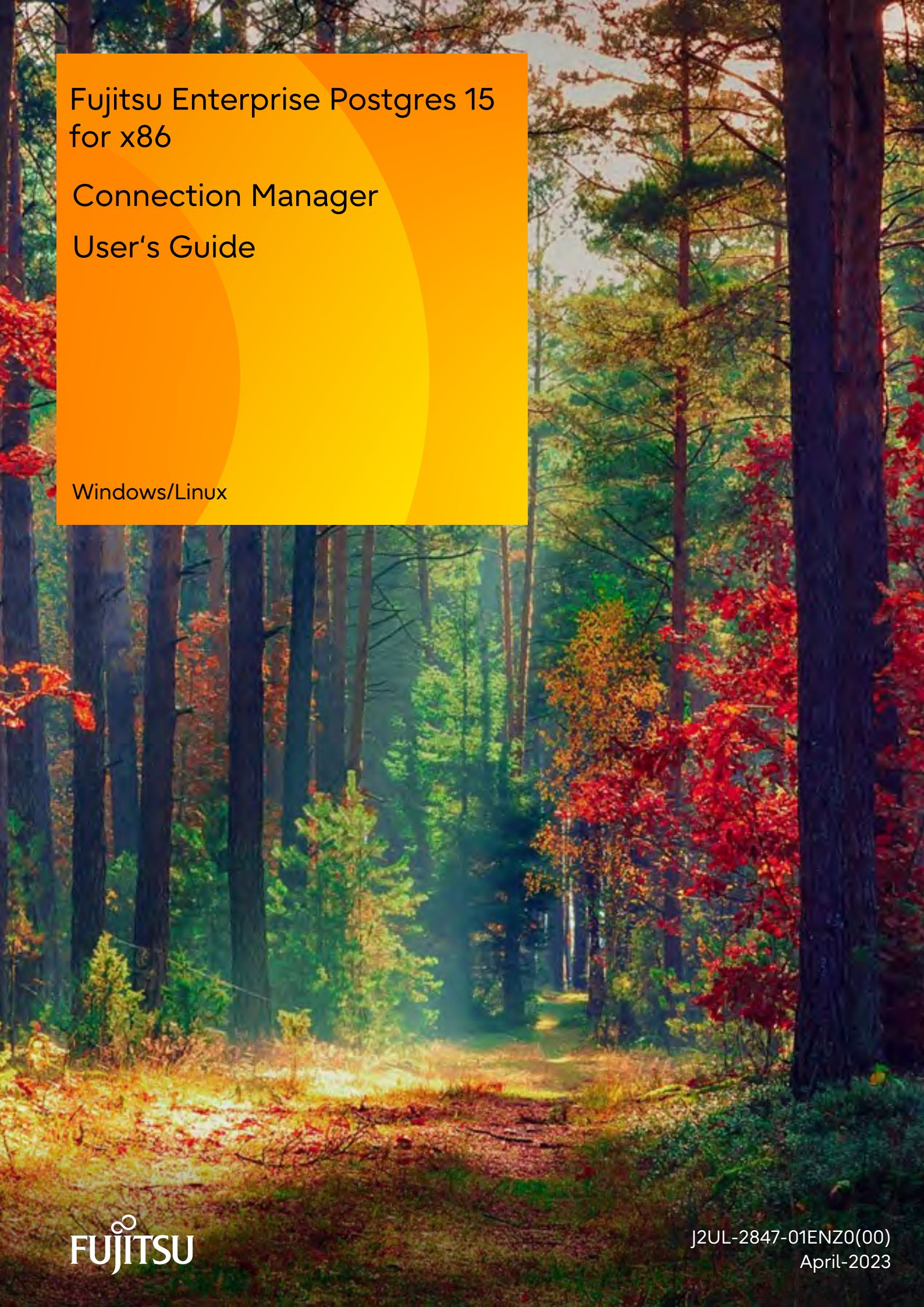
Unmounting the File System..... 12

Fujitsu Enterprise Postgres 15 for x86

Connection Manager

User's Guide

Windows/Linux



FUJITSU

J2UL-2847-01ENZ0(00)
April-2023

Preface

Purpose of this document

This document describes the Connection Manager features of Fujitsu Enterprise Postgres.

Intended readers

This document is aimed at people who use the Connection Manager features.

Readers of this document are also assumed to have general knowledge of:

- Fujitsu Enterprise Postgres
- PostgreSQL
- Linux
- Windows

Structure of this document

This document is structured as follows:

Chapter 1 Connection Manager Features

Explains the features and Mechanisms of the Connection Manager.

Chapter 2 Setting Up

Explains setting up the Connection Manager.

Chapter 3 Using from an Application

Explains how to use the Connection Manager from an application.

Appendix A System Views

Explains the system view of Connection Manager.

Export restrictions

If this document is to be exported or provided overseas, confirm legal requirements for the Foreign Exchange and Foreign Trade Act as well as other laws and regulations, including U.S. Export Administration Regulations, and follow the required procedures.

Issue date and version

Edition 1.0: April 2023

Copyright

Copyright 2020-2023 Fujitsu Limited

Contents

Chapter 1 Connection Manager Features 1
1.1 Heartbeat Monitoring Feature 1
1.1.1 Difference from TCP keepalive 1
1.1.2 Mechanism of Heartbeat Monitoring Feature 2
1.2 Transparent Connection Support Feature 3
1.2.1 Mechanism of Connections using Transparent Connection Support Feature 3

Chapter 2 Setting Up 5
2.1 Setting Up the Client Side 5
2.1.1 Setup Preparation Tasks 5
2.1.1.1 Security Policy Settings 5
2.1.1.2 Preparing for Output to the Event Log 5
2.1.2 Creating a Directory for the conmgr Process 6
2.1.3 Configuring conmgr.conf 6
2.1.4 Registering the conmgr Process with a Windows Service 10
2.2 Setting Up the Server Side 11
2.2.1 Configuring postgresql.conf 11
2.2.2 Introducing the watchdog extension 12
2.2.3 Windows Firewall Settings 12
2.3 Starting the conmgr Process 13
2.3.1 Starting and Stopping the conmgr Process 13
2.3.2 Setting of Automatic Start / Stop of conmgr Process 14
2.4 Removing Setup 14

Chapter 3 Using from an Application 16
3.1 Connection Method 16
3.2 How to Detect Instance Errors 16
3.3 How to Use in libpq 16
3.3.1 How to Specify Multiple Connection Destinations 17
3.3.2 Using the Asynchronous Connection Method 17
3.3.3 Using an Asynchronous Communication Method 17
3.3.4 Behavior of PQhost() or PQhostaddr() or PQport() 17
3.3.5 Behavior of PQstatus() 17
3.3.6 PQcmSocket() 18
3.4 How to Use in ODBC Driver 18
3.4.1 Behavior of SQLGetInfo() 18
3.5 How to Use in JDBC Driver 18
3.5.1 Behavior of loadBalanceHosts Parameter 18

Appendix A System Views 19
A.1 pgx_stat_watchdog 19

Index 20

Chapter 1 Connection Manager Features

The Connection Manager provides the following features:

Heartbeat monitoring feature

Detects kernel panics between the server running the client and the server running the PostgreSQL instance(hereinafter referred to as instance), physical server failures, and inter-server network link downs, and notifies the client or instance. The client is notified as an error event through the SQL connection, and the instance will be notified in the form of a force collection of SQL connections with clients that are out of service.

Transparent connection support feature

When an application wants to connect to an instance of an attribute in a set of instances configured for replication, it can connect to that instance without being aware of which server it is running on.

Information

The available client drivers for Connection Manager are libpq (C language library), ECPG (embedded SQL in C), ECOBPG (embedded SQL in COBOL), JDBC driver and ODBC driver.

Each function is described below.

1.1 Heartbeat Monitoring Feature

Describes the Connection Manager's heartbeat monitoring feature.

Note

The Connection Manager does not monitor for delays, such as CPU busy occurring in the postmaster process or in the backend processes to which the application connects directly, or for no response, such as due to a software bugs. It also does not monitor application downtime or unresponsiveness. To detect these, use various timeout features provided by PostgreSQL or the client drivers.

1.1.1 Difference from TCP keepalive

A peer of TCP connections cannot automatically detect a link down or server down.

There are two main methods to detect it. One is the operating system (Not all operating systems support it) TCP keepalive feature, and the other is the keepalive-equivalent timeout function implemented at the application layer. Connection Manager's heartbeat monitoring capabilities are categorized as the latter.

The operating system TCP keepalive feature has the following disadvantages, but the Connection Manager's heartbeat monitoring feature does not:

- The keepalive does not work when the TCP layer cannot receive an acknowledgement (ACK) and retransmits the packet repeatedly. This means that it is not possible to detect a down (For example, if a network goes down,) before sending some data and receiving ACK from the other side. There is also a parameter to interrupt retransmissions, which is not supported by some operating systems. The Connection Manager's heartbeat monitoring feature does not have this disadvantage because it is timeout monitoring at the application layer.
- The periodic packets for keepalive are sent per-TCP socket. If a instance accepts too many (For example, a few thousand clients) SQL connections, the load on the instance side cannot be ignored. The Connection Manager's heartbeat monitoring feature greatly reduces the load by allowing packets to be sent to the instance on a per-server basis on which the client runs.

1.1.2 Mechanism of Heartbeat Monitoring Feature

On the client side, the user must start one monitoring process using the cm_ctl command for the set of the instances to be monitored. This process, called the "conmgr process", can only be started by a user who is not an administrator (e.g. superuser(root) on Linux), or by a user who does not have Windows administrator privileges. An instance set is a collection of one or more instances that make up replication. One configuration file (conmgr.conf) for each conmgr process is used to set the information about the set of the instances being monitored and the parameters for monitoring.

On the server side, by installing PostgreSQL's EXTENSION that is called "watchdog", the postmaster will start two processes as background workers at instance startup.

One is the process for sending and receiving packets to and from the conmgr process for heartbeat monitoring. It is called "watchdog process". The other is the process for forcibly terminating SQL connections of the clients for which the watchdog process detects a failure on heartbeat monitorting. It is called "terminator process". SQL connections that do not use Connection Manager is also terminated, because the terminator process terminates them by IP address as key.

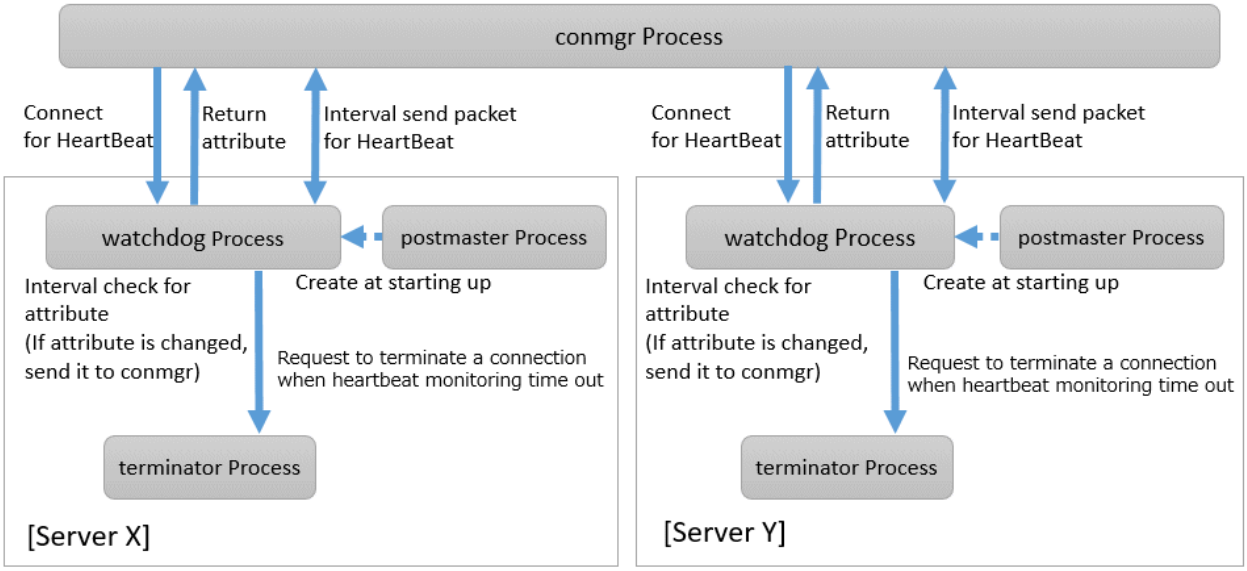
Note

System Configuration Notes

For replication, it is recommended that the instance that connects to the upstream instance of replication and the conmgr process that regards the upstream instance as an instance to be monitored for heartbeart (specified in backend_host parameter or backend_hostaddr parameter that is a configuration parameter of conmgr process) be not placed on the same server. This is because if the conmgr process stops normally or abnormally, the terminator process in the upstream instance will also kill the replication connection. The replication connection will reconnect automatically even if it is forcibly disconnected, so replication will continue without any problems. However, this can be a problem when the replication load is high or on systems that are sensitive to replication delays.

Note that the replication connection have different monitoring feature than the Connection Manager, so there is no need to monitor the Connection Manager for heartbeat. Refer to PostgreSQL documentation for details.

The process relationship is as follows:



See

Refer to "cm_ctl" in the Reference for information on cm_ctl command.

1.2 Transparent Connection Support Feature

The features similar to Connection Manager's transparent connection support feature can be found in PostgreSQL's libpq and other client drivers.

Using libpq as an example, the connection parameter to use that feature is target_session_attrs parameter. If this parameter is used not through Connection Manager, libpq will attempt to find the required instance by connecting sequentially to all instances of the set of instance requested by the host parameter or hostaddr parameter. In the worst case, libpq may find the promoted primary at the connection to the last instance of instance set. This means that you cannot predict how long it will take to complete the switch.

However, when conmbined with the Connection Manager, the conmgr process obtains its attributes via the watchdog process from all servers in a set of servers in advance, so that the connections to that server can be initiated as soon as the application requests it.

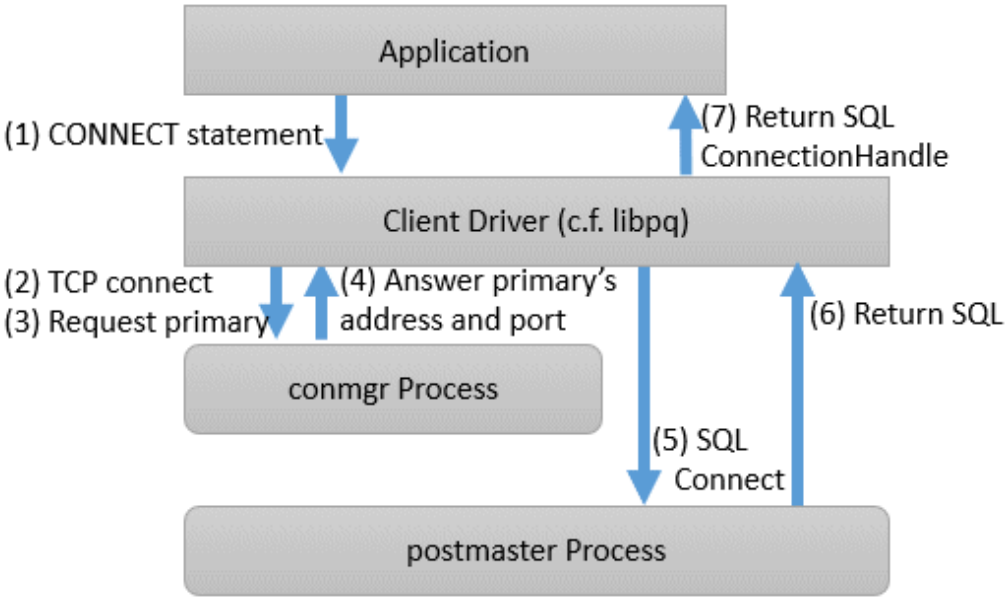
1.2.1 Mechanism of Connections using Transparent Connection Support Feature

A connection using this mechanism actually consists of two steps, but from the perspective of the application, it looks like a single SQL connection. In the application's connection string, specify the IP address or host name (In most cases it is "localhost") and port number where the conmgr process listens, and target_session_attrs parameter. You do not need to explicitly state that the connection is to the conmgr process. This is because the client driver can automatically determine whether the connection is to a instance or a conmgr process.

In the first phase of the connection, the client driver receives a connection request from the application and connects to the location specified in the connection string. Initially, it uses the protocol PostgreSQL requests, and if it learns in the middle that the connection is to a conmgr process, it asks the conmgr process for the IP address and port number that the instance with the attributes specified in the connection parameter target_session_attrs is listening for. If the destination is a backend process rather than a conmgr process, the connection process completes immediately and continues to send and receive data for normal SQL execution. The first stage of processing falls within the scope of timeout monitoring for SQL connection processing by each client driver. For example, the connection_timeout parameter of libpq.

In the second phase of the connection, the client driver connects to the instance using the IP address and port number from the conmgr process.Thereafter, the client driver and the instance directly send and receive the data for SQL execution. This ensures that the Connection Manager does not affect the performance of the SQL execution.

When the client driver is waiting to receive data after the second stage is completed, it monitors the reception of data to the two sockets obtained at each stage of the connection. This allows the client driver to know when, for example, the conmgr process notifies the client of a network link down.


Application
(1) CONNECT statement
(7) Return SQL
ConnectionHandle
Client Driver (c.f. libpq)
(2) TCP connect
(3) Request primary
(4) Answer primary's
address and port
(6) Return SQL
conmgr Process
(5) SQL
Connect
postmaster Process

Chapter 2 Setting Up

Describes setting up the Connection Manager.

2.1 Setting Up the Client Side

On the client side, configure settings for the conmgr process.

2.1.1 Setup Preparation Tasks

Describes the preparatory tasks to perform before you set up the conmgr process.

2.1.1.1 Security Policy Settings

In order for the Windows service to start and stop the conmgr process, Connection Manager requires a security setting that allows the OS user account that is to start the conmgr process to log on as a service.

Describes how to set up security so that you can log on as a service.

1. Displaying the [Local Security Policy] window

   In Windows, select [Administrative Tools], and then click [Local Security Policy].

2. Setting up security

   1. In the [Local Security Policy] window, select [Security Settings], select [Local Policies], and then click [User Rights Assignment].
   2. Under [Policy] in the [User Rights Assignment] window, double-click [Log on as a service].
   3. In the [Log on as a service Properties] window, set the following:

      a. Select the [Local Security Setting] tab.

      b. On the [Local Security Setting] tab, click [Add User or Group].

      c. In the [Select Users or Groups] window, enter the operating system user account of the instance administrator user in [Enter the object names to select].

      d. Click [OK].

   4. In the [Log on as a service Properties] window, click [OK].

2.1.1.2 Preparing for Output to the Event Log

Describes how to prepare for outputting error logs to the event log.

Note

If you have not registered the event source name, the content of the message output to the event log might be incomplete.

Register this default event source name before using the cm_ctl command because the event log may be output with a default event source name of "conmgr".

The following is an example of registering a DLL for a 64 bit product as the event source name "conmgr".

```
> regsvr32 /n /i:"conmgr" "c:\Program Files\Fujitsu\fsepv<x>client64\lib\pgevent.dll"
```

For a multi-version installation

If Fujitsu Enterprise Postgres is already installed on the same machine, note the path to the registered DLLs by searching for the following key in the Registry Editor. Then, register a new DLL with the default event source name.

The DLL path is used to re-register the default event source name during removing setup.

```
conmgr
```

2.1.2 Creating a Directory for the conmgr Process

You need one conmgr process for each set of instances that you want to configure for replication. Assign a dedicated directory to each conmgr process. This directory must assign read, execute, and write permissions for the user who starts the conmgr process.

This directory is specified when you run the cm_ctl command, which starts and stops the conmgr process. To specify a directory in the cm_ctl command, set it in the environment variable CMDATA or specify it in the -D option.

See

Refer to "cm_ctl" in the Reference for information on cm_ctl.

2.1.3 Configuring conmgr.conf

Place the configuration file conmgr.conf in the directory for the conmgr process.

Syntax for conmgr.conf

- In conmgr.conf, after the symbol(#) are considered comments.
- The parameter name = value" is a set of settings and must be written on one line.
- Set the value in a format that matches the type of each parameter. The types and formats are:
  - integer: Numeric type. Express as a sequence of numbers in decimal number.
  - string: String type. You can also include spaces by enclosing them in quotation marks('). If you include quotation marks, escape them.
  - enum: Enumeration type. Possible values are determined.

Parameters to Set

port (integer)

Specify the port number on which the conmgr process listens for connections from the applications.

The value must be greater than or equal to 1 and less than or equal to 65535. The default is 27546. You must restart conmgr process for this parameter change to take effect.

backend_host* (string)

Specify the host name or IP address of the instance.

You can also use IPv6 address. If you specify the IP address directly, you can save time by using backend_hostaddr parameter. If backend_host parameter and backend_hostaddr parameter are both specified, backend_hostaddr parameter is used. You must restart conmgr process for this parameter change to take effect.

To distinguish multiple instances, append a zero-based number immediately after the parameter name, such as backend_host0, backend_host1,... This number is called the instance number. A parameter identified by the same instance number configures the settings of a single instance. If you want to exclude some instances from your replication configuration, you can simply remove the settings for that instance.

Note

Refer to "System Configuration Notes" in "1.1.2 Mechanism of Heartbeat Monitoring Feature" for details.

Note

If the primary is not included in the instances configured in conmgr.conf, use the -W option when starting Connection Manager with the cm_ctl command. Without the -W option, the cm_ctl command will not return until the primary connection is complete.

For example, if two instances are listening on "host name:host0, port number:5432" and "host name:host1, port number: 2345", write as follows.

```
backend_host0='host0'
backend_port0=5432
backend_host1='host1'
backend_port1=2345
```

You can also mix different instance number settings:

```
backend_host0='host0'
backend_host1='host1'
backend_port0=5432
backend_port1=2345
```

It does not matter if the instance number is missing as in the following (instance number 1):

```
backend_host0='host0'
backend_host2='host2'
backend_port0=5432
backend_port2=2345
```

If the host name is omitted, as in instance number 1 below, an error will occur when loading the configuration file.

```
backend_host0='host0'
backend_host2='host2'
backend_port0=5432
backend_port1=5555
backend_port2=2345
```

backend_hostaddr*(string)

Same as backend_host parameter except no name resolution is used.

backend_port* (integer)

Specify the port number the postmaster of the instance will listen on.

The value must be greater than or equal to 1 and less than or equal to 65535. The default is 27500. Append the instance number as you would for backend_host parameter. You must restart conmgr process for this parameter change to take effect.

watchdog_port* (integer)

Specify the port number on which the watchdog process listens.

The conmgr process connects to this port, but the user application does not. you must set it to the same value as watchdog.port parameter in postgresql.conf. The value must be greater than or equal to 1 and less than or equal to 65535. The default is 27545. Append the instance number as you would for backend_host parameter. You must restart conmgr process for this parameter change to take effect.

heartbeat_interval (integer)

Specify the interval at which heartbeat packets are sent for heartbeat monitoring.

Used in conjunction with heartbeat_timeout parameter. Connection Manager heartbeat monitoring always continues to send packets periodically from both ends of the connection. If a packet is not received from the other side within a certain period of time, the link is considered down.

Note that this method is different from TCP keepalive. TCP keepalive send a keepalive packet only when there is a certain amount of inactivity (idle), and expects to receive an ACK for that packet. If TCP keepalive does not receive an ACK, it repeats this a specified number of times and then assumes that the link is down.

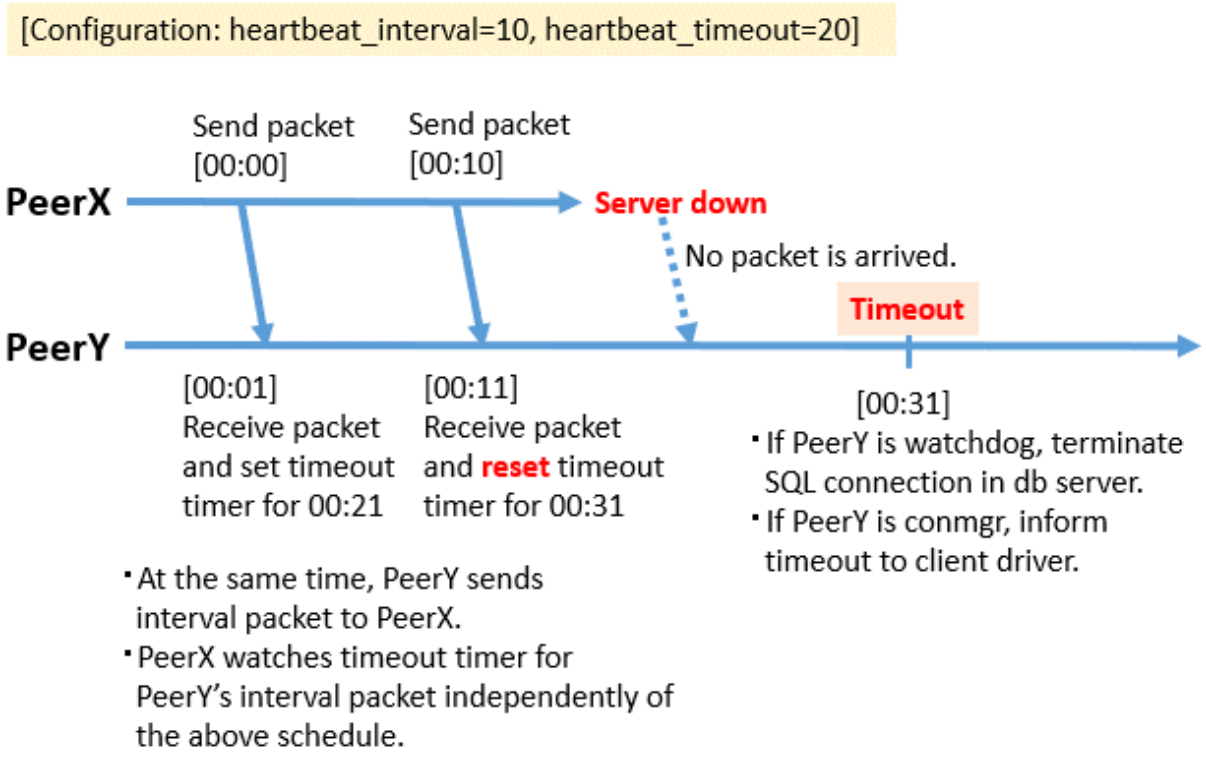
The heartbeat_interval parameter and heartbeat_timeout parameter are propagated from the conmgr process to the watchdog process, and also apply to the interval between the transmissions of heartbeat packets from the watchdog process. If a watchdog process is connected from both a conmgr process with a heartbeat_interval parameter of 3 seconds and a conmgr process with a heartbeat_interval of 5 seconds, it sends heartbeat packets every 3 seconds to the former process and every 5 seconds to the latter process. The unit is seconds. Specify a value equal to or more than 1 second. The default is 10 seconds. You must restart conmgr process for this parameter change to take effect.

heartbeat_timeout (integer)

If a heartbeat packet for heartbeat monitoring cannot be received for more than the time specified by this parameter, an error is assumed to have occurred and the application is notified of the error.

This parameter should be decide of heartbeat_interval parameter as the basis. No error is occured when the configuration file is loaded, but is always considered abnormal by heartbeat monitoring if it is at least not greater than heartbeat_interval parameter. The unit is seconds. Specify a value equal to or more than 1 second. The default is 20 seconds. You must restart conmgr process for this parameter change to take effect.

Refer to the following figure for the relationship between the heartbeat_interval parameter and heartbeat_timeout parameter settings and the heartbeat timeout.



heartbeat_connect_interval (integer)

Specify the interval between attempts to establish heartbeat monitoring again after detecting an abnormality.

This parameter is useful when only the database server is started, but not the instance. In such a situation, the TCP connection fails immediately, and retries cannot be attempted without an interval. If you specify an excessively long value, you may delay noticing the start of the instance. If a connection attempt fails for a long time, it will attempt the next connection after the time specified by heartbeat_connect_interval parameter has elapsed. The unit is seconds. Specify a value equal to or more than 1 second. The default is 1 second. You must restart conmgr for this parameter change to take effect.

heartbeat_connect_timeout (integer)

Specify the connection timeout for establishing heartbeat monitoring.

The connection includes the time it takes to send the TCP connection and the first heartbeat packet to the watchdog process and receive a reply from the watchdog process. This parameter is particularly needed when the other server is down or the network is disconnected. This is because TCP connections are attempted over a long period of time, depending on the operating system configuration, and the connection takes a long time to fail. The unit is seconds. Specify a value equal to or more than 1 second. The default is 10 seconds. You must restart conmgr process for this parameter change to take effect.

log_destination (string)

Specify the destination of the message.

You can specify multiple destinations. Use commas to separate multiple entries and enclose all in single quotation marks.

L

"stderr" and "syslog" can be specified. The default is to print only to stderr. You must restart conmgr process for this parameter change to take effect.

W

"stderr" and "eventlog" can be specified. The default is to print only to stderr. You must restart conmgr process for this parameter change to take effect.

L

syslog_facility (enum)

Specify the syslog facility.

Valid only if log_destination parameter includes "syslog".

LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, or LOCAL7 can be specified. The default is "LOCAL0". You must restart conmgr process for this parameter change to take effect.

L

syslog_ident (string)

Specify the program name used to identify the output from the conmgr process.

The default is "conmgr". You must restart conmgr process for this parameter change to take effect.

W

event_source (string)

Specify the event source name used to identify the output from the conmgr process.

The default is "conmgr". You must restart conmgr process for this parameter change to take effect.

log_min_messages (enum)

Specifies the level of messages to output.

It can be DEBUG, INFO, NOTICE, WARNING, ERROR, LOG, FATAL, or PANIC. Messages below the specified level are not output. The default is "WARNING". You must restart conmgr process for this parameter change to take effect.

max_connections (integer)

Specifies the maximum number of simultaneous connections to the conmgr process.

If there are more than this maximum number of client connections, it forces the connection to be closed without sending an error message to the client.

The conmgr process also outputs this fact at level “LOG” to the destination specified by log_destination. Specify a value equal to or more than 0.

If 0 is specified, there is no limit. The default is 0. You must restart conmgr process for this parameter change to take effect.

L

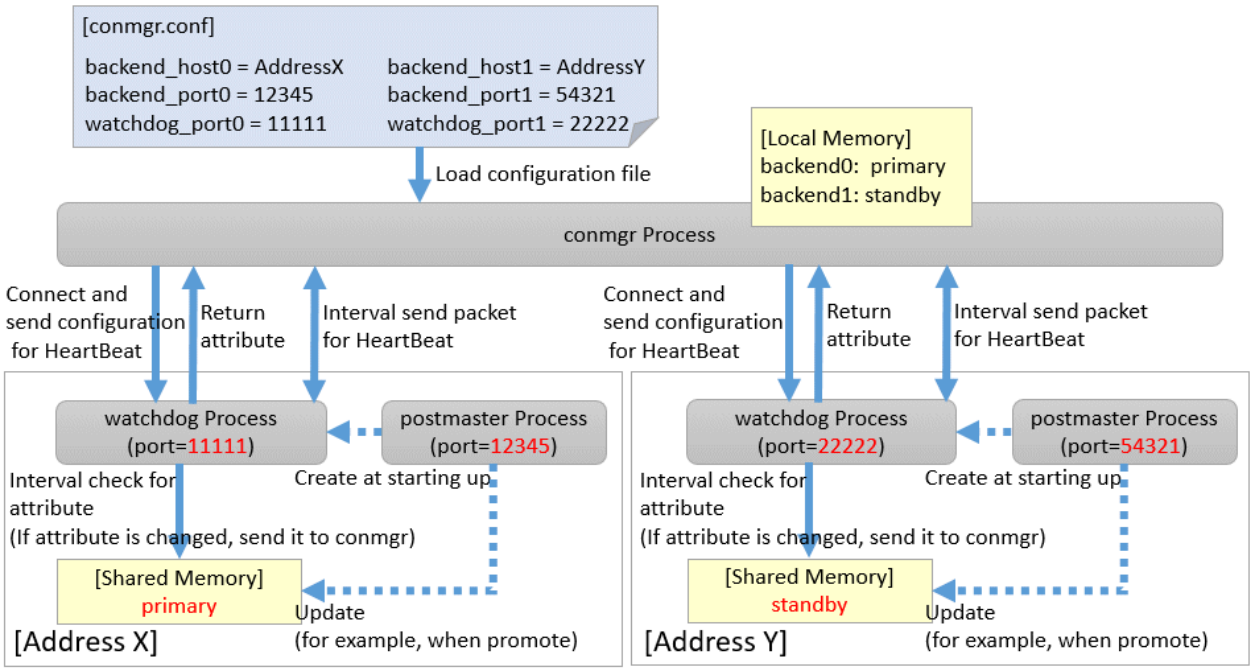
Note

The maximum number of file descriptors that can be opened simultaneously (You can check it with -n of the ulimit command.) imposed on a conmgr process by the OS user limit should be greater than the value derived from the following equation:. Otherwise, the conmgr process will abort if the user limit is violated.

```
9 + Number of database instances specified in conmgr.conf x 2  + max_connections
specified in conmgr.conf
```

Overview of connections definitions

The following figure shows the relationship between the IP address or host name and the port number set in conmgr.conf and the processes.



2.1.4 Registering the conmgr Process with a Windows Service

To start or stop the conmgr process using a Windows service, register the conmgr process with the Windows service. Describes how to register the conmgr process with a Windows service.

Register the conmgr process with a Windows service by using the cm_ctl command with the service name to register, the user name, the password, and the path to the directory where the conmgr.conf is located.

Example)

The following example is executed when the registration service name is "conmgr", the user name is "fsepuser", and the directory where confgr.conf is located is "D: \conmgr".

```
> cm_ctl register -N "conmgr" -U fsepuser -P ******** -D D:\conmgr
```

Note

- Execute the command as a Connection Manager administrator user with "Administrator" privileges. Execute the command from the [Administrator: Command Prompt] window. Right-click [Command Prompt], and then select [Run as administrator] from the menu to display the [Administrator: Command Prompt] window.
- Do not specify the --complete option when registering Windows services in register mode of the cm_ctl command. This is because the conmgr process may be running in the background if it fails to start using Windows services.

- You must specify a user name and password for the following reasons:

  Because Windows services are started under the Network Service account, the conmgr process is created for that account. As a result, the conmgr process may fail to start on another user. For security reasons, if you do not specify a user name and password, specify the account from the list of Windows services immediately after you register the conmgr process with the Windows service.

- When entering the password for the cm_ctl command, be careful that it is not referenced by other users for security reasons.

2.2 Setting Up the Server Side

On the server side, configure settings for the watchdog process.

2.2.1 Configuring postgresql.conf

Describes the postgresql.conf parameters that must be set when using the Connection Manager.

Parameters to Set

max_connections

An existing PostgreSQL parameter. Add 2 to the value already set.

Connection to the instance is maintained from the time the instance is started to do the following:

- The watchdog process checks the state of the instance.
- The terminator process forces the client to terminate the SQL connection.

shared_preload_libraries

An existing PostgreSQL parameter. Add a watchdog.

The watchdog process and terminator process start when you add watchdog and restart the instance.

watchdog.port (integer)

Specify the port number on which the watchdog process accepts connections for heartbeat monitoring from the conmgr process.

The value must be greater than or equal to 1 and less than or equal to 65535. The default is 27545. The instance must be restarted for this parameter change to take effect.

watchdog.check_attr_interval (integer)

Specify the interval between checking the attributes of a instance.

watchdog process immediately notifies the conmgr process if the attribute changes.

The unit is milliseconds. Specify a value equal to or more than 1 millisecond. The default is 1000 milliseconds. The instance must be restarted for this parameter change to take effect.

watchdog.max_heartbeat_connections (integer)

Specify the maximum number of conmgr processes that connect to watchdog process.

The default is the value specified in max_connections of postgresql.conf.

There is no upper limit, but about 200 bytes of memory are consumed for 1 connection when PostgreSQL is started.

L

Note

Normally you do not need to consider, but if you have a heartbeat connection with a very large number of conmgr processes, it may violate on the maximum number of file descriptors (You can check it with -n of the ulimit command.)

of the OS user limit. This is because the socket for the heartbeat connection consumes the file descriptor. Set the maximum number of file descriptors of the OS user limit to a value larger than the value calculated below from the max_files_per_process parameter value and watchdog.max_heartbeat_connections parameter value in postgresql.conf.

```
max_files_per_process + watchdog.max_heartbeat_connections x 2
```

2.2.2 Introducing the watchdog extension

Execute the CREATE EXTENSION statement with watchdog.

Example)

```
postgres=# CREATE EXTENSION watchdog;
CREATE EXTENSION
```

This allows you to see the pgx_stat_watchdog view for information about the watchdog process.

2.2.3 Windows Firewall Settings

Describes how to enable the port number used by the watchdog process when you enable the Windows Firewall feature.

Windows Server(R) 2016:

1. In the [Windows Firewall] window, click [Advanced settings] on the left side of the window.
2. In the [Windows Firewall with Advanced Security] window, click [Inbound Rules] on the left side of the window.
3. Click [New Rule] on the right side of the window.
4. In the [New Inbound Rule Wizard] window, select [Port], and then click [Next].
5. Select [TCP] and [Specific local ports], then specify the "port number on which the watchdog process accepts connections for life-and-death monitoring from the conmgr process (watchdog.port)" defined in postgresql.conf in the text box, and then click [Next].
6. Select [Allow the connection], and then click [Next].
7. Select the profiles for which this rule applies, and then click [Next].
8. In [Name], specify the desired name(ex: Connection Manager), and then click [Finish].
9. In the [Windows Firewall with Advanced Security] window, check if the added rule is enabled under [Inbound Rules] in the center of the window.

Other than above:

1. In the [Windows Defender Firewall] window, click [Advanced settings] on the left side of the window.
2. In the [Windows Defender Firewall with Advanced Security] window, click [Inbound Rules] on the left side of the window.
3. Click [New Rule] on the right side of the window.
4. In the [New Inbound Rule Wizard] window, select [Port], and then click [Next].
5. Select [TCP] and [Specific local ports], then specify the "port number on which the watchdog process accepts connections for life-and-death monitoring from the conmgr process (watchdog.port)" defined in postgresql.conf in the text box, and then click [Next].
6. Select [Allow the connection], and then click [Next].
7. Select the profiles for which this rule applies, and then click [Next].

8. In [Name], specify the desired name(ex: Connection Manager), and then click [Finish].
9. In the [Windows Defender Firewall with Advanced Security] window, check if the added rule is enabled under [Inbound Rules] in the center of the window.

2.3 Starting the conmgr Process

2.3.1 Starting and Stopping the conmgr Process

Describes starting and stopping the conmgr process.

You can use commands related to Windows services to start, stop, and check the health of the conmgr process.

To use a Windows service, register the confmgr process with the Windows service.

Note

Although it is possible to start and stop the conmgr process by running the cm_ctl command without registering the conmgr process with the Windows service, we recommend that you use the Windows service to start and stop the conmgr process for the following reasons.

The cm_ctl command starts the conmgr process as a user process. Therefore, closing the command prompt window from which the command was executed will force the conmgr process to stop.

Starting the conmgr process

Start with the net start or sc start command and the service name.

You can also start it from the Windows Services window by following the steps below.

1. Display the [Services] window

   In [Administrative Tools], click [Services].

2. Start a service

   From the list of service names, select the conmgr process and click the [Start Service] button.

Stopping the conmgr process

Start with the net stop or sc stop command and the service name.

You can also stop it from the Windows Services window by following the steps below.

1. Display the [Services] window

   In [Administrative Tools], click [Services].

2. Stop a service

   From the list of service names, select the conmgr process and click the [Stop Service] button.

Checking the operating status of the conmgr process

To check whether a service is running immediately after the start operation of the conmgr process, check the status of the service as follows.

1. Display the [Services] window

   In [Administrative Tools], click [Services].

2. Check the status of the service

   Check the service status of the conmgr process from the list of service names.

Use the cm_ctl command to check the operating status of the conmgr process during operation.

See

Refer to "cm_ctl" in the Reference for information on cm_ctl.

2.3.2 Setting of Automatic Start / Stop of conmgr Process

The conmgr process can be started/stopped automatically according to the start/stop of the database server OS.

If the service startup type is "manual", change it to "automatic". By setting it to "automatic", it will start automatically when the database server OS is started, and will automatically stop when the database server OS is shut down.

Note

Please do so as a user with "Administrator" privileges.

Follow the procedure below to switch services.

1. Display the [Services] window

   In [Administrative Tools], click [Services].

2. Switch startup

   Select the service name of the conmgr process, display the [Properties] dialog, and switch the startup type from Manual to Automatic.

You can also change the above settings with the sc config command.

2.4 Removing Setup

Describes how to removing setup the Connection Manager.

Linux

No work is required on the client side.

On the server side, drop watchdog extension by DROP EXTENSION statement and remove it from shared_preload_libraries.

Example)

```
postgres=# DROP EXTENSION watchdog;
DROP EXTENSION
```

Windows

The following work is required on the client side.

1. Unregister Windows service

   Cancel the conmgr process registered in the Windows service.

   In unregister mode of the cm_ctl command, specify the registered service name to cancel the conmgr process of the Windows service.

Example)

An execution example is shown when the registered service name is "conmgr".

```
> cm_ctl unregister -N "conmgr"
```

Note

Execute the command as a Connection Manager administrator user with "Administrator" privileges. Execute the command from the [Administrator: Command Prompt] window. Right-click [Command Prompt], and then select [Run as administrator] from the menu to display the [Administrator: Command Prompt] window.

2. Deletion of registration for event log

   If you are outputting to the event log in "2.1.1.2 Preparing for Output to the Event Log", delete the registered event source name.

   Example)

```
> regsvr32 /u /i:"conmgr" "c:\Program Files\Fujitsu\fsepv<x>server64\lib\pgevent.dll"
```

For a multi-version installation

If the conmgr process created using this package is configured to output the error log to the event log, re-register the default event source name using the DLL pathname that was saved in "2.1.1.2 Preparing for Output to the Event Log".

On the server side, use the DROP EXTENSION statement to remove the watchdog extension and remove the watchdog from shared_preload_libraries.

Example)

```
postgres=# DROP EXTENSION watchdog;
DROP EXTENSION
```

Chapter 3 Using from an Application

Describes how to use the Connection Manager from an application.

3.1 Connection Method

When connecting to the instance using ConnectionManager, specify the following values in the connection parameters of the application. Application connection parameters are parameters that specify the database IP address, host name, port number, etc., which are originally specified when connecting to the database from the application. For example, when using libpq, specify "localhost" for the host parameter and specify the port number on which the conmgr process listens for the port parameter.

Connection parameters not shown here are used directly by the instance in the second stage of the connection, connecting to the instance (connecting to an instance without the Connection Manager), and the conmgr process does not check or use it.

Connection destination address

Specify "localhost". Unix domain sockets are not allowed.

It is possible to connect to a remote conmgr process, but it should not be used for other purposes expect such as testing. This is because there is no mechanism between the application and the conmgr process to detect the remote server down or the network link down, making the Connection Manager meaningless.

Port number

Specify the value specified for the port parameter in conmgr.conf.

Connection destination instance attributes

Follow the "Target server" in the application connection switch feature. Refer to "Taget server" in "Connection Information for the Application Connection Switch Feature" in the "Application Development Guide" for information on the target server in the application connection switch feature.

Note

Connection Manager cannot be used if preferPrimary is specified for targetServerType when using the JDBC driver.

3.2 How to Detect Instance Errors

Only special if you are using libpq's asynchronous communication method. For additional discovery methods, refer to "Errors when an Application Connection Switch Occurs and Corresponding Actions" of the for each client driver in the "Application Development Guide" . If the conmgr process goes down while accessing it, or if the conmgr process tries to establish a SQL connection while it is down, the same error is returned as if the instance went down.

3.3 How to Use in libpq

libpq provides very detailed communication control. Therefore, to detect a heartbeat error through the conmgr process, you may need to modify the existing application logic.

See

Refer to "libpq - C Library" in the PostgreSQL Documentation on functions described below.

3.3.1 How to Specify Multiple Connection Destinations

The host parameter or hostaddr parameter in the connection string not only specifies the destination of one conmgr process, but can also be a mixture of destinations of other conmgr processes and postmaster. In this case, the connections are tried in the order listed.

For example, if the connection string specifies conmgr1, conmgr2 in that order, and if conmgr1 does not know the server for the attribute specified in target_session_attrs parameter, it queries conmgr2 for the destination. And, for example, if postmaster1, conmgr1 is specified, it will attempt to connect directly to the database instance pointed to by postmaster1. If this fails, query conmgr1 for a connection.

3.3.2 Using the Asynchronous Connection Method

An asynchronous connection method is to use a function like PQconnectStart() instead of a function like PQconnectdb (). PQconnectStart() returns without synchronizing the completion of the connection to the database. The user application must then monitor the sockets returned by PQsocket() to be readable or writable, for example by using the poll() system call, according to the values required by the return value of PQconnectPoll().

With the Connection Manager, the socket returned by PQsocket() may change after a call to PQconnectPoll(), so be sure to reacquire the socket that you want to give to the poll() system call using PQsocket(). This behavior is similar to simply specifying multiple hosts in the connection string without using the Connection Manager.

3.3.3 Using an Asynchronous Communication Method

An asynchronous communication method is one in which the application returns control without waiting for a response from the database, and PQsetnonblocking() is used to asynchronize completion of transmission or completion of receipt of all results. Instead of using a function like PQexec(), use a function like PQsendQuery(). In this method, the user application monitors the socket that connects to the database returned by PQsocket(), for example, by using the poll() system call.

For example, if the link to the database goes down, simply monitoring the socket returned by PQsocket with the poll() system call will not detect it.

However, it is possible to detect the reception of database anomaly detection packets sent from the conmgr process, for example, by monitoring the reception of data on the socket (POLLIN) connecting to the conmgr process returned by PQcmSocket(). Once a reception is detected, the user application need not directly manipulate the packet. By calling something like PQgetResult() or PQcosumeInput() according to the existing application logic, it behaves as if the connection were disconnected. Refer to "Errors when an Application Connection Switch Occurs and Corresponding Actions" in the Application Development Guide on SQLSTATE returned,etc. If you are not using the Connection Manager, PQcmSocket() returns -1.

3.3.4 Behavior of PQhost() or PQhostaddr() or PQport()

PQhost(), PQhostaddr() or PQport() typically return a host parameter or hostaddr parameter or port parameter specified in the connection string by the user application. However, if you specify a connection destination for the conmgr process, the destination for the conmgr process you specify will be changed to the database connection destination information provided by the conmgr process before the connection is completed. This behavior is similar to simply specifying multiple hosts in the connection string without using the Connection Manager.

3.3.5 Behavior of PQstatus()

If you are using an asynchronous connection method, you can monitor the intermediate state of the connection to the database with PQstatus(). If you are using the Connection Manager, the enum value returned by PQstatus() is appended with the following:

```
CONNECTION_AWAITING_CMRESPONSE
/ * Waiting for a response from the conmgr process * /
```

3.3.6 PQcmSocket()

You can get a socket that leads to the conmgr process. It returns a value equal to or more than 0 for a valid socket, or -1 if you are not connected to the conmgr process.

```
int PQcmSocket(const PGconn *conn);
```

3.4 How to Use in ODBC Driver

Describes points to note when using the Connection Manager using the ODBC driver.

3.4.1 Behavior of SQLGetInfo()

When SQL_SERVER_NAME is specified in the argument InfoType, SQLGetInfo () normally returns the contents set in Servername or Server of the data source. However, if you specify a connection destination for the conmgr process, the destination for the conmgr process you specify will be changed to the database connection destination information provided by the conmgr process before the connection is completed. This behavior is similar to simply specifying multiple hosts in the connection string without using the Connection Manager.

3.5 How to Use in JDBC Driver

Describes points to note when using the Connection Manager using the JDBC driver.

3.5.1 Behavior of loadBalanceHosts Parameter

The loadBalanceHosts parameter is a connection parameter for the JDBC driver to use the load balancing feature. You can specify whether to use the load balancing feature by setting this parameter. However, Connection Manager provides a unique load balancing feature that users cannot specify whether to use or not. Therefore, even if the user sets the loadBalanceHosts parameter to disable the JDBC driver load balancing feature, the Connection Manager load balancing feature is always enabled when connecting to the database via the Connection Manager.

Appendix A System Views

A.1 pgx_stat_watchdog

A row in this view corresponds to conmgr process, which is connected to watchdog process. Additional columns may be added in future versions versions.

Column	Type	Description
conmgr_addr	inet	IP address of conmgr process.
conmgr_port	integer	The conmgr (ephemeral) port number that conmgr process is using to communicate with watchdog process. This is not the port number to be set in conmgr.conf.
heartbeat_interval	integer	The interval at which heartbeat packets are sent to and from this conmgr process. The unit is seconds.
heartbeat_timeout	integer	The timeout value for the heartbeat to and from this conmgr process. The unit is seconds.

Index

[B]

backend_host* 6
backend_hostaddr* 7
backend_port* 7

[C]

conmgr.conf 6
conmgr process 2

[E]

event_source 9

[H]

Heartbeat monitoring feature 1
heartbeat_connect_interval 8
heartbeat_connect_timeout 8
heartbeat_interval 7
heartbeat_timeout 8

[L]

log_destination 9
log_min_messages 9

[M]

max_connections (integer) 9
max_connections 11

[P]

pgx_stat_watchdog 19
port 6
postgresql.conf 11
PQcmSocket() 18

[S]

shared_preload_libraries 11
syslog_facility 9
syslog_ident 9

[T]

terminator process 2
Transparent connection support feature 1

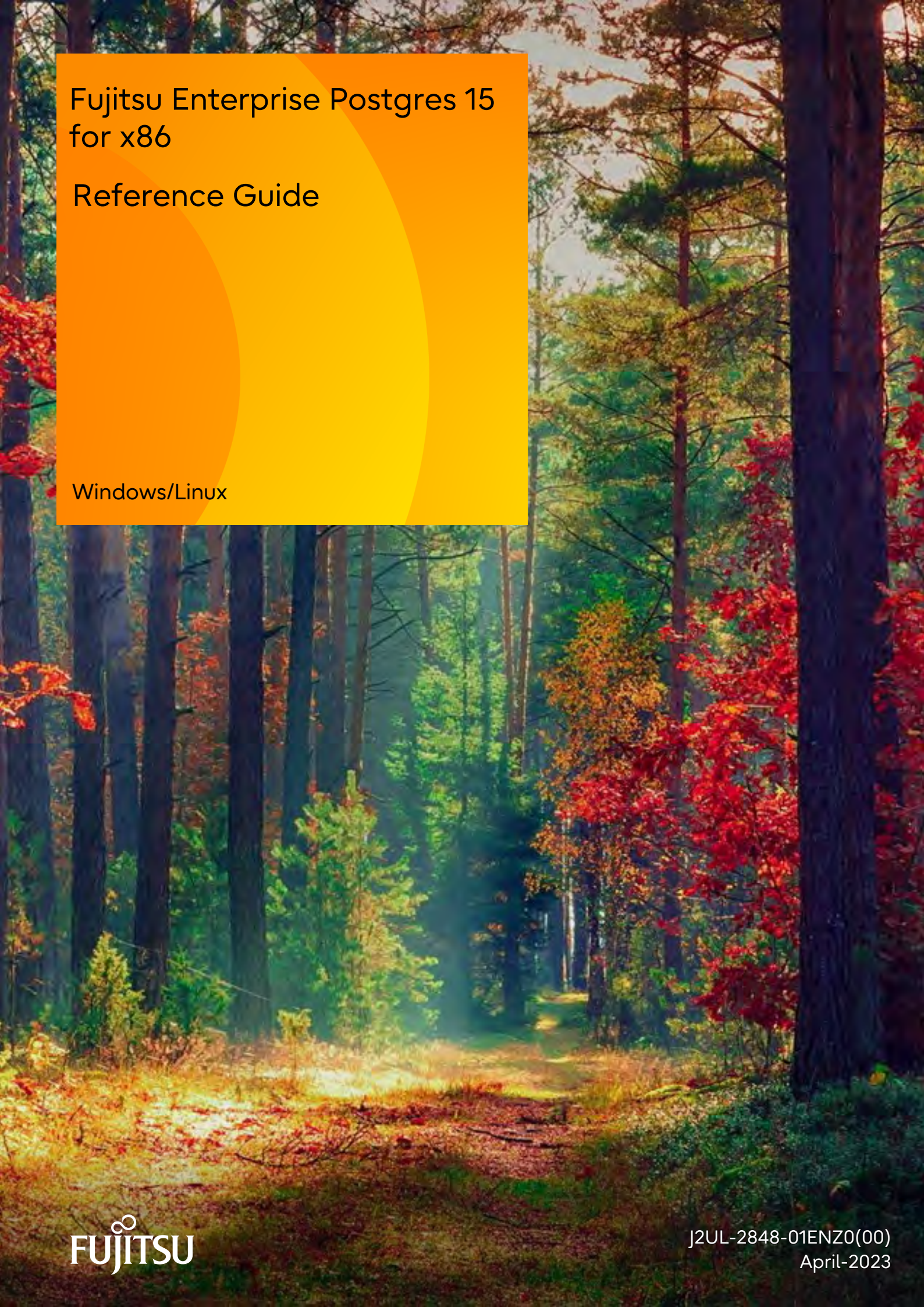
[W]

watchdog.check_attr_interval 11
watchdog.max_heartbeat_connections 11
watchdog.port 11
watchdog process 2
watchdog_port* 7

Fujitsu Enterprise Postgres 15 for x86

Reference Guide

Windows/Linux



FUJITSU

J2UL-2848-01ENZ0(00)
April-2023

Preface

Purpose of this document

This document is a command reference, and explains Fujitsu Enterprise Postgres commands and options with features expanded on from PostgreSQL.

Intended readers

This document is aimed at people who manage and operate Fujitsu Enterprise Postgres. Readers of this document are also assumed to have general knowledge of:

- PostgreSQL
- SQL
- Linux
- Windows

Structure of this document

This document is structured as follows:

Chapter 1 Command List and Specification Format

Lists commands and describes their specification format.

Chapter 2 Client Commands

Explains options not listed in "PostgreSQL Client Applications" in the PostgreSQL Documentation.

Chapter 3 Server Commands

Explains commands and options not listed in "PostgreSQL Server Applications" in the PostgreSQL Documentation.

Chapter 4 Mirroring Controller Commands

Explains the Mirroring Controller commands.

Chapter 5 Connection Manager Commands

Explains the Connection Manager commands.

How to read this document

Examples in this document are predominantly for UNIX/Linux.

For Windows, replace values (such as paths in the examples) as appropriate.

Export restrictions

If this document is to be exported or provided overseas, confirm legal requirements for the Foreign Exchange and Foreign Trade Act as well as other laws and regulations, including U.S. Export Administration Regulations, and follow the required procedures.

Issue date and version


```
Edition 1.0: April 2023
```


Copyright

Copyright 2015-2023 Fujitsu Limited

Contents

Chapter 1 Command List and Specification Format....1
1.1 Command List....1
1.1.1 Client Commands....1
1.1.2 Server Commands....1
1.1.3 Mirroring Controller Commands....1
1.1.4 Connection Manager Commands....2
1.2 Command Specification Format....2

Chapter 2 Client Commands....3
2.1 pg_dumpall....3
2.2 pgx_loader....3

Chapter 3 Server Commands....6
3.1 pg_ctl....6
3.2 pgx_dmpall....6
3.3 pgx_fjqssinf....8
3.4 pgx_keystore....9
3.5 pgx_rcvall....11
3.6 postgres....13

Chapter 4 Mirroring Controller Commands....15
4.1 mc_arb....15
4.2 mc_ctl....17

Chapter 5 Connection Manager Commands....22
5.1 cm_ctl....22

Chapter 1 Command List and Specification Format

This chapter lists commands and describes their specification format.

1.1 Command List

This chapter lists commands and options not explained in "PostgreSQL Client Applications" or in "PostgreSQL Server Applications" in the PostgreSQL Documentation.

1.1.1 Client Commands

The commands below have options not explained in "PostgreSQL Client Applications" in the PostgreSQL Documentation.

Command	Functional overview
pg_dumpall	Extract a PostgreSQL database cluster into a script file

The command below is not explained in "Client Applications" in the PostgreSQL Documentation.

Command	Functional overview
pgx_loader	Loads data from an external file into a PostgreSQL table.

1.1.2 Server Commands

The commands below have options not explained in "PostgreSQL Server Applications" in the PostgreSQL Documentation.

Command	Functional overview
pg_ctl	Initialize, start, stop, or control a PostgreSQL server
postgres	PostgreSQL database server

The commands below are not explained in "PostgreSQL Server Applications" in the PostgreSQL Documentation.

Command	Functional overview
pgx_dmpall	Backs up the data directory, tablespaces, and configuration files.
L pgx_fjqssinf	Collects failure investigation data (SLES 15 only)
pgx_keystore	Manages keystore
pgx_rcvall	Recovers the data directory, tablespaces, and configuration files.

1.1.3 Mirroring Controller Commands

Mirroring Controller has the following commands:

Command	Functional overview
mc_arb	L Start, stop and display the status of the Mirroring Controller arbitration process.

Command	Functional overview
	W Start, stop and display the status of the Mirroring Controller arbitration process, or register and unregister as a Windows service.
mc_ctl	L Start and stop Mirroring Controller, switch/ disconnect the server, or display the server status. W Start and stop Mirroring Controller, switch/ disconnect the server, display the server status, or register and unregister as a Windows service.

1.1.4 Connection Manager Commands

Connection Manager has the following commands:

Command	Functional overview
cm_ctl	Start, stop or display the status of the conmgr process

1.2 Command Specification Format

The table below shows the command specification format.

Item	Explanation
[ ]	Indicates optional element.
...	Indicates that the item can be specified repeatedly.

Chapter 2 Client Commands

This chapter explains options not listed in "PostgreSQL Client Applications" in the PostgreSQL Documentation.

2.1 pg_dumpall

Name

pg_dumpall -- Extract a PostgreSQL database cluster into a script file

Synopsis

```
pg_dumpall [connectionOption...] [option...]
```

Options

--no-tablespace-encryption

Do not output commands to encrypt tablespaces. Running the generated SQL script will restore the originally encrypted data without being encrypted.

See

Refer to "pg_dumpall" in the PostgreSQL Documentation for details.

2.2 pgx_loader

Name

pgx_loader --Loads data from a file into a PostgreSQL table.

Overview

```
pgx_loader load -c command [options...]

pgx_loader recovery -t table
```

Description

The pgx_loader command loads data from an external file into PostgreSQL tables, and commits or rolls back transactions prepared during data load.

In load mode, data is loaded at high speed by executing the COPY FROM command specified in *command* at a certain degree of parallelism. If load is completed successfully, the message below is output to the standard output.

```
LOAD count
Note: count is the number of rows loaded.
```

Refer the pgx_stat_progress_loader view to see the progress of the load process.

In recovery mode, commit or rollback of transactions prepared during data load is performed.

Refer to "pgx_stat_progress_loader" in the Operation Guide for pgx_stat_progress_loader view.

Options

-a
--echo-sql

Display the executed command in the standard output.

-c *command*
--copy-command=*command*

Specify the COPY FROM command to be executed. If STDIN is specified for the FROM clause, data will be loaded from the standard input. In this case, SQL superuser privileges (or having one of the roles pg_read_server_files or pg_execute_server_program) are not required, because local user access privileges will be used for external files and external programs, instead of database server access privileges.

"binary" cannot be specified for the FORMAT option of the COPY FROM command specified in this option. Therefore, input files in binary format cannot be specified.

The FREEZE option cannot be specified for the COPY FROM command specified in this option.

See

Refer to "COPY" in the PostgreSQL Documentation for information on the COPY FROM command.

-j *number-of-jobs*
--jobs=*number-of-jobs*

Specify the number of background workers (parallel workers) that the COPY COMMAND should use to simultaneously perform data conversion, table creation, and index creation. This option can dramatically reduce the time for loading a large amount of data on instances that runs on multiple processor machines.

The optimal value depends on the server, client, and network configurations. The number of CPU cores and disk configuration also affect the optimal value. The number of CPU cores of the server is recommended as the initial value to try. Naturally, if a value that is too large is used, performance degradation will occur due to thrashing and context switching.

Specify a value from 2 to 128. The default is 2.

-t *table*
--table=*table*

Complete the prepared transactions only for the specified table.

-?
--help

Show how to use pgx_loader command line arguments, and exit.

The command line options below control the database connection parameters.

-d *connstr*
--dbname=*connstr*

Specify the database name to connect to.

If this option is not specified, the PGDATABASE environment variable will be used. If the environment variable is not set, your operating-system user name will be used.

-h *host*
--host=*host*

Specify the host name of the machine the database server runs on. If the specified value starts with a slash, it will be used as the directory for a Unix domain socket.

If this option is not specified, the PGHOST environment variable will be used. If the environment variable is not set, it will be considered a Unix domain socket connection.

-p *port*
--port=*port*

Specify the TCP port to be used by the server to monitor the connection, or extension of the local Unix domain socket file.

If this option is not specified, the PGPORT environment variable will be use. If the environment variables is not set, 27500 will be used.

-U *username*
--username=*username*

User name for connection to the database.

-w
--no-password

Never prompt for the password. If the server requires password authentication but other means (such as a .pgpass file) are not available, the connection attempt will fail. This option can be useful in batch jobs, scripts, and so on, where no user is present to enter a password.

-W
--password

Force pgx_loader to prompt for the password before connecting to the database. This option is never essential, since pgx_loader will automatically prompt for the password if the server demands password authentication. However, pgx_loader will waste a connection attempt finding out if the server requires a password. In some cases it is worth specifying this option to avoid the extra connection attempt.

Diagnostics

load mode

0: Normal exit

Other than 0: Abnormal exit

recovery mode

0: There are no prepared transactions that must be completed

3: A prepared transaction was committed

4: A prepared transaction was rolled back

Other than the above: Abnormal exit

The order of the table rows loaded by pgx_loader may not match the order of the file lines. This is because the file lines will have been inserted into the table in parallel, by multiple parallel workers.

Example

The example below loads the file /path/to/data.csv (2000 records) into table tbl using a degree of parallelism of 3.

```
$ pgx_loader load -j 3 -c "COPY tbl FROM '/path/to/data.csv' WITH CSV"
LOAD 2000
```

The example below reads the file /path/to/data.csv (2000 records) from the standard input and loads into table tbl using a degree of parallelism of 3.

```
$ pgx_loader load -j 3 -c "COPY tbl FROM STDIN WITH CSV" < /path/to/data.csv
LOAD 2000
```

The example below completes the transactions prepared for table tbl.

```
$ pgx_loader recovery -t tbl
```

Chapter 3 Server Commands

This chapter explains commands and options not listed in "PostgreSQL Server Applications" in the PostgreSQL Documentation.

3.1 pg_ctl

Name

pg_ctl -- Initialize, start, stop, or control a PostgreSQL server

Synopsis

```
pg_ctl start [-D datadir] [-l filename] [-W] [-t seconds] [-s]
           [-o options] [-p path] [-c] [--keystore-passphrase | --kms-secret]

pg_ctl restart [-D datadir] [-m s[mart] | f[ast] | i[mmediate] ]
             [-W] [-t seconds] [-s] [-o options] [-c]
             [--keystore-passphrase | --kms-secret]
```

Options

--keystore-passphrase

Prompt for the passphrase to open the keystore when using a file-based keystore.

--kms-secret

When you use a key management system as a keystore, you are prompted to enter secret information to open the keystore.

See

Refer to "pg_ctl" in the PostgreSQL Documentation for details.

3.2 pgx_dmpall

Name

pgx_dmpall - Backs up the data directory, tablespaces, and configuration files.

Synopsis

```
pgx_dmpall  [option...]
```

Description

The pgx_dmpall command backs up the data directory, tablespaces, and configuration files. The backup data is stored in the directory specified by backup_destination parameter of postgresql.conf. The pgx_dmpall command also deletes archived Write Ahead Logs (WAL) that are no longer necessary for recovery when the backup completes successfully.

Options

-c

This option only backs up configuration files. The configuration files are as follows:

- postgresql.conf (postgresql.conf)
- File for host-based authentication (pg_hba.conf)
- Configuration file for ident authentication (pg_ident.conf)

If an external reference, such as 'include' in postgresql.conf, is set, the reference destination files are also backed up.

-C fast|spread
--checkpoint=fast|spread

Sets checkpoint mode to fast or spread (default).
If fast is specified, the checkpoint processing at the start of backup becomes quick, but the impact on performance of running applications gets larger due to intense I/O. In spread mode, the impact on applications is smaller but the backup takes longer, because the checkpoint is performed slowly.

-D *datadir*

Specify the data directory. If this option is omitted, the value of the environment variable PGDATA is used.

-f *configFile*

Specify the postgresql.conf configuration file. This option is set if the data directory and the configuration file set in the 'data_directory' parameter of the postgresql.conf file are running in separate directories.

-P *tablespacesListFile*
--tablespaces-list-file=*tablespacesListFile*

Specify the full path of the file containing the names of the tablespaces to be backed up using the copy command, using less than 1024 bytes.

The file format is described below:

tablespaceName<newline>
tablespaceName<newline>
...

Tablespaces not listed in the specified file are backed up to the backup storage directory. If this option is not specified, all tablespaces are backed up using the copy command.
This option can be specified if the -Y option has been specified, and it is used to limit the tablespaces backed up using the copy command.

-U *username*
--username=*username*

Specify the user name of the database superuser. This defaults to the name of the effective user running pgx_dmpall.

-Y *copyCommandFile*
--copy-command=*copyCommandFile*

Specify the full path of the file of the copy command for backup, using less than 1024 bytes. This option cannot be specified together with the -c option.

-w
--no-password

Never issue a password prompt. If the server requires password authentication and a password is not available by other means such as a .pgpass file, the connection attempt will fail. This option can be useful in batch jobs and scripts where no user is present to enter a password.

-W
--password

Force pgx_dmpall to prompt for a password before connecting to a database.

This option is never essential, since pgx_dmpall will automatically prompt for a password if the server demands password authentication. However, pgx_dmpall will waste a connection attempt finding out that the server wants a password. In some cases it is worth typing -W to avoid the extra connection attempt.

--maintenance-db=*dbname*

Specifies the name of the database to connect to. If not specified, the postgres database will be used; if that does not exist, template1 will be used.

Any database can be specified as long as it can be connected to.

--exclude-copy-cluster

Excludes a database cluster from backup via the copy command. This option can be specified if the -Y option has been specified. If this option is not specified, the database cluster will be backed up using the copy command.

Environment

PGDATA

Specify the data directory. You can overwrite using the -D option.

Diagnostics

0: Normal end

Other than 0: Abnormal end

Notes

This command can only be executed when the database server is running.

Execute this command as a PostgreSQL user account.

Do not update or delete files in the backup storage directory. Otherwise, you may not be able to recover the database.

Do not store other files in the backup storage directory.

This command uses one database connection. To establish a connection, this command uses the IPv4 loopback address 127.0.0.1 on Windows, and the UNIX domain socket on other operating systems. Therefore, permit these connections in pg_hba.conf.

This command cannot be executed on the standby server.

Example

In the following example, the data directory, tablespaces, and configuration files are backed up. At this time, stored WALs are no longer necessary because the backups are destroyed.

```
$ pgx_dmpall
```

Related item

pgx_rcvall

3.3 pgx_fjqssinf

Name

pgx_fjqssinf - Collects failure investigation data.

Synopsis

```
pgx_fjqssinf  -i {1|2|3} [-w directory]
```

Description

When the cause of a trouble that occurred during the construction or operation of the environment is not identified, information for the initial investigation is collected. The collected investigation information is created in the destination directory as pgx_fjqssinf_YYYYMMDD_HHMMSS/.

Options

-i {1|2|3}

Specifies the incident of the trouble that occurred. 1 is specified for for process error, 2 for process result error, and 3 for no response. This option must be specified for the database server to gather information. Specify 1 when collecting information with the arbitration server.

-w directory

Specifies the destination directory for the collected data. The default is /tmp.

Environment (When the information is collected by a database server)

PGDATA

Specifies the data directory.

PGDATABASE

Specifies the name of the database to connect to. Any database that can be connected to may be specified.

PGPORT

Specifies the port number of the instance. This should not be specified if the default port number (27500) has not been changed.

PGUSER

Specifies the user name of the database superuser. The database superuser must be configured to allow client authentication.

MCCONTROLDIR

Specifies the Mirroring Controller management directory. This should only be specified when database multiplexing operation is set up.

Environment (When information is collected by an arbitration server)

ARBCONTROLDIR

Specifies the management directory for the Mirroring Controller arbitration process.

ARBUSER

Specifies the OS user that initialized the Mirroring Controller arbitration process.

Diagnostics

0: Normal end

Other than 0: Abnormal end

Notes

This command must be executed as a superuser (root) account.

Example

Below is an example of collecting information for initial investigation in the event of a process failure.

```
# pgx_fjqssinf -i 1
```

Below is an example of collecting information for an initial investigation by an arbitration server.

```
# pgx_fjqssinf -i 1
```

3.4 pgx_keystore

Name

pgx_keystore -- Manages keystore

Synopsis

```
pgx_keystore [-a|--enable-auto-open] [option...] keystore_location

pgx_keystore [-s|--obfuscate-secret] [option...]
```

Description

pgx_keystore enables auto-open of a keystore.

Options

To enable automatic opening using a file-based keystore

-a
--enable-auto-open

Enables auto-open of a keystore. This allows the keystore to open automatically without entering the passphrase when the database server starts.

When auto-open is enabled, an obfuscated copy keystore.aks is created in the same directory where the keystore file keystore.ks is stored. To disable auto-open, delete keystore.aks.

-P *passphrase*
--passphrase=*passphrase*

Specify the passphrase to open the keystore. If this option is omitted, the prompt to enter the passphrase is displayed.

keystore_location

Specify the absolute or relative path of the keystore file.

To enable automatic opening using a key management system as the keystore

-s

--obfuscate-secret

Obfuscates the secret information needed to connect to the key management system. By specifying obfuscated private information as an authentication option in the key management system connection information file, the keystore is opened automatically when the database server is started without entering the key management system credentials.

--secret = *secret*

Specify the secret information required to connect to the key management system. If you omit this option, you are prompted for the secret.

-o *obfuscated-secret-file*

Specifies the file that contains the obfuscated secret. If the file already exists, the command terminates abnormally without overwriting it.

Diagnostics

0: Normal exit

Other than 0: Abnormal exit

Notes

This command can be executed whether the database server is running or stopped. It does not connect to the database server.

This command does not connect to the key management system.

Example

Enables automatic keystore opening when using a file-based keystore.

```
$ pgx_keystore -a /key/store/location/keystore.ks
```

Enable automatic keystore opening by obfuscating sensitive credentials when using the key management system as a keystore.

```
$ pgx_keystore -s -o /example/keypassphrase.ksc
Enter secret:
```

3.5 pgx_rcvall

Name

pgx_rcvall - Recovers the data directory, tablespaces, and configuration files.

Synopsis

```
pgx_rcvall [option...]
```

Description

The pgx_rcvall command recovers the data directory, tablespaces, and configuration files using the data that was backed up with pgx_dmpall command and archived Write-Ahead-Log (WAL). If none of the options that indicate the recovery point is specified, all archived WAL are applied and the data will be recovered to the latest point.

Options

-B *backupdir*

Specify the backup storage directory. If the data directory is damaged, this option cannot be omitted.

-D *datadir*

Specify the data directory. If this option is omitted, the value of the environment variable PGDATA is used.

-e *targetTime*

Specify this option to recover the data as of the specified date and time.

targetTime

Specify the time at which the data is recovered. The format is as follows:

```
"YYYY-MM-DD HH:MM:SS"
```

-l

This option displays a list of the backup data information in the backup storage directory that was obtained using the pgx_dmpall command. If the pgx_dmpall command was executed using the copy command (-Y option) for backup, the resources backed up using the copy command will also be listed. This cannot be specified together with -p, -e or -n option.

-n *restorePoint*

Specify this option to recover the data to the specified restore point. Restore points are created with SQL function pg_create_restore_point. If multiple restore points with the same names were created, the first one after the backup was taken is used for recovery. If the specified restore point does not exist, the recovery fails. This cannot be specified together with -e or -p option.

-p

Specify this option to recover the data as of the time when the last backup completed. This cannot be specified together with -e or -n option.

-x

Specify this option if you do not want to include transactions committed at the time specified in the -e option as part of the recovery.

-Y *copyCommandFile*
--copy-command=*copyCommandFile*

Specify the full path of the file of the copy command for recovery, using less than 1024 bytes. This option cannot be specified together with the -l option.

--keystore-passphrase

Prompt for the passphrase to open the keystore when using a file-based keystore.

--kms-secret

When you use a key management system as a keystore, you are prompted to enter secret information to open the keystore.

--view-results-of-copying

Output the backup information file that was written by the copy command executed via the pgx_dmpall command. This option cannot be specified together with the -l, -p, -e, -n, or -Y option.

Environment

PGDATA

Specify the data directory. You can overwrite using the -D option.

PGPORT

Specify the port number for connecting to the database.

PGUSER

Specify the user name of the database superuser. This defaults to the name of the effective user running pgx_dmpall.

Diagnosis

0: Normal exit

Other than 0: Abnormal exit

Backup data information

Date

Date the backup data was created using the pgx_dmpall command.

Dir

This is the name of the directory in the backup storage directory that is used to store the backup data.
Directory naming format: Time format (YYYY-MM-DD_HH-MM-SS)

Status

This is the status of the pgx_dmpall command backup data.

COMPLETE: Complete

INCOMPLETE: Incomplete

Resources backed up by the copy command

List of resources that were backed up by the copy command executed via the pgx_dmpall command.

If there are resources that were backed up by the copy command, then database clusters ('pg_data') or tablespace names will be listed, delimited by header and halfwidth comma.

Notes

This command can only be executed when the database server is stopped, except when it is executed with -l option.

Execute this command as a PostgreSQL user account.

Use backup data that was taken from the recovery target data directory.

Before executing this command, disconnect all application database connections. Additionally, do not connect to the database during recovery.

The configuration files are restored from those files that were taken by the last pgx_dmpall (including -c option).

This command connects to the database to determine whether the recovery has completed. So ensure that you set the port number with PGPORT environment variable in the environment where multiple instances exist.

Match the OS timezone setting when running pgx_dmpall/pgx_rcvall to the timezone specified by timezone parameter in postgresql.conf.

Otherwise, data might be recovered to an unexpected time when -e or -p is specified.

If you recover to a past point, a new timeline (history of database updates) begins at that point. That recovery point is the latest point in the new timeline when the recovery is completed. If you subsequently recover to the latest point, the database updates in the new timeline will be replayed.

Valid restore points are the ones that were created in the timeline where the backup had been taken. That means that if you recover to a past point, those restore points created thereafter are unavailable. Therefore, take a backup when you have restored the past data desired.

If the pgx_dmpall command was executed using the copy command (-Y option) for backup, it is necessary to execute this command using the copy command (-Y option) for recovery. However, because the list of resources (database cluster or tablespace) that were backed up using the copy command is recorded in the backup directory, there is no need to specify the target resources when executing this command. The -I option can be used to check the target resources for which a backup is retrieved using the copy command.

Example

In the following example, the data directory, tablespaces, and configuration files are recovered.

```
$ pgx_rcvall -B /home/pgsql/Backupdir
```

In the following example, the data directory and tablespaces are recovered at 10:00:00 on 01-02-2022. The configuration files are recovered at the point at which the last of the data is obtained.

```
$ pgx_rcvall -B /home/pgsql/Backupdir -e "2022-02-01 10:00:00"
```

In the following example, the data directory and tablespaces are recovered up to the time of restore point "before_match_20220210_1". The configuration files are restored from the latest backup.

```
$ pgx_rcvall -B /home/pgsql/Backupdir -n before_match_20220210_1
```

In the following example, the obtained backup data information in the backup storage directory is displayed in a list.

```
$ pgx_rcvall -l
```

Related item

pgx_dmpall

3.6 postgres

Name

postgres -- PostgreSQL database server

Synopsis

```
postgres [option...]
```

Options

-K

Prompt for the passphrase to open the keystore. This option works in single-user mode only, so you must also specify the --single option, as shown below:

postgres --single -K

See

Refer to "postgres" in the PostgreSQL Documentation for details.

Chapter 4 Mirroring Controller Commands

This chapter explains the Mirroring Controller commands.

4.1 mc_arb

Name

L mc_arb - Start, stop, and display the status of the Mirroring Controller arbitration process

W mc_arb - Start, stop, and display the status of the Mirroring Controller arbitration process, or register and unregister it as a Windows service

Overview

```
mc_arb start [-M mcdir] [-w| -W]

mc_arb stop [-M mcdir] [-e]

mc_arb status [-M mcdir]
```

W
```
mc_arb register [-M mcdir] [-w| -W] -P password [-S a[uto] | d[emand]]
mc_arb unregister [-M mcdir]
```

Description

L mc_arb starts, stops, and displays the status of the Mirroring Controller arbitration process.

W mc_arb starts, stops, and displays the status of the Mirroring Controller arbitration process, or registers and unregisters it as a Windows service.

The start mode starts the Mirroring Controller arbitration process.

The stop mode stops the Mirroring Controller arbitration process.

The status mode displays the connection status of the Mirroring Controller arbitration process with the Mirroring Controller processes of the primary server and standby server.

W The register mode registers the Mirroring Controller arbitration process as a Windows service. The -w and -W options can be used to start/stop the Mirroring Controller arbitration process through the Windows service, and when it executes the mc_arb command, the specified option will take effect.

The unregister mode unregisters the Mirroring Controller arbitration process as a Windows service.

If the Mirroring Controller arbitration process has not been started on the server executing the command, stop mode and status mode will terminate with an error.

Additionally, if Mirroring Controller is forcibly stopped on the database server, it may take a few moments until the status mode displays the status of the server connection as offline.

L This command can be executed by any user.

W Execute this command as a user with administrator privileges (operating system user that belongs to the "Administrators" group).

Options

-e

Specify this option to forcibly stop the Mirroring Controller arbitration process on the active server.

Specify this option to stop the Mirroring Controller arbitration process but keep Mirroring Controller running (to stop both, first stop Mirroring Controller of the database server, and then the Mirroring Controller arbitration process). This option can also be specified to halt the arbitration process (by stopping the Mirroring Controller arbitration process) when the fencing command called by the arbitration process becomes unresponsive.

-M *mcdir*

Specify the Mirroring Controller arbitration process management directory.

ASCII characters can be specified in the directory path.

If this option is omitted, the value of the environment variable ARBCONTROLDIR is used.

W -P *password*

For the register mode, specify the password for the user who executed the command.

W -S a[uto] | d[emand]

Specify the start type for the Windows service to be registered. You can choose auto or demand as the start type by specifying the entire word or just its first letter. The default is auto.

-w

Waits for operations to finish.

L This option is the default of start mode.

W This option is the default of start mode and register mode.

-W

Does not wait for operations to finish.

Environment

ARBCONTROLDIR

Specify the Mirroring Controller arbitration process management directory.

ASCII characters can be specified in the directory path.

You can specify the -M option to override this value.

Diagnostics

0: Normal end

Other than 0: Abnormal end

Notes

If the Mirroring Controller arbitration process is forcibly stopped or communication between the command and the Mirroring Controller arbitration process is interrupted while the Mirroring Controller arbitration process is being stopped, a message that the command is being executed may be output and stopping may terminate in error, even though no other instances of the mc_arb command are being executed.
To solve this issue, ensure that other instances of the mc_arb command are not being executed before forcibly stopping the Mirroring Controller arbitration process.

W Use the start mode and stop mode to start and stop the Windows service. Before being able to use these modes, use the register mode to register the Mirroring Controller arbitration process as a Windows service.
The mc_arb command must be executed from "Administrator: Command Prompt". Right-click [Command Prompt], and then select [Run as administrator] from the context menu to display the [Administrator: Command Prompt] window.

Example

L Start the Mirroring Controller arbitration process.

```
$ mc_arb start -M /mcarb_dir/arbiter1
```

Start the Mirroring Controller arbitration process.

```
> mc_arb start -M D:\mcarb_dir\arbiter1
```

Display details of mc_arb status

```
server_id   host            status
(1)         (2)             (3)
```

```
(1) Server identifier
(2) Host name or IP address
(3) Server connection status
    online   : Connected
    offline  : Disconnected
```

4.2 mc_ctl

Name

mc_ctl - Start and stop Mirroring Controller, switch/disconnect the server, or display the server status.

mc_ctl - Start and stop Mirroring Controller, switch/disconnect the server, display the server status, or register and unregister as a Windows service.

Overview

```
mc_ctl start [-M mcdir] [-w| -W] [-f| -F] [--mc-only] [--async-connect-arbiter] [--local-server server_id]

mc_ctl stop [-M mcdir] [[-a] [--mc-only]| [-e][--local-server server_id]]

mc_ctl status [-M mcdir] [--arbiter] [--local-server server_id]

mc_ctl switch [-M mcdir] [--force [--no-fencing ]] [--local-server server_id]

mc_ctl detach [-M mcdir] [--no-fencing] [--local-server server_id]

mc_ctl enable-failover [-M mcdir] [--local-server server_id]

mc_ctl disable-failover [-M mcdir] [--local-server server_id]
```

```
mc_ctl register [-M mcdir] [-w| -W] [-f| -F] [--mc-only] -P password
                [-S a[uto] | d[emand]] [--async-connect-arbiter] [--local-server server_id]
```

```
mc_ctl unregister [-M mcdir] [--local-server server_id]
```

Description

mc_ctl starts and stops Mirroring Controller, switches/disconnects the server, or displays the server status.

mc_ctl starts and stops Mirroring Controller, switches/disconnects the server, displays the server status, or registers and unregisters as a Windows service.

The start mode starts Mirroring Controller. If the --mc-only option is omitted, the command starts a database instance.

The stop mode stops Mirroring Controller. If the --mc-only option is omitted, the database instance is stopped. If --mc-only option is not specified, database instance is also stopped. When executes on standby server without --mc-only, standby server will be detached from primary server.

The status mode displays the status of the servers, database instance processes, and disks monitored by Mirroring Controller. Additionally, if the --arbiter option is specified, Mirroring Controller arbitration process connection status is displayed.

The switch mode switches the primary server. When the server is switched, the database instance on the primary server stops, and the database instance on the standby server is upgraded to primary server and begins degrading operation. This mode can be executed on the primary server and standby server in an environment where the Mirroring Controller process can communicate with the Mirroring Controller process on the other server.

The detach mode forcibly disconnects the standby server. This mode is used to forcibly disconnect the other server when stopping of Mirroring Controller cannot be performed using stop mode (which requires login to the standby server). It can only be executed on the primary server.

The enable-failover mode enables automatic switching and disconnection.

The disable-failover mode disables automatic switching and disconnection.

W

The register mode registers Mirroring Controller as a Windows service. The -w, -W, -f, -F, --mc-only, --async-connect-arbiter, and --local-server options are used when Mirroring Controller is started and stopped as a Windows service. If the mc_ctl command is used to start and stop Mirroring Controller, the option specified in the command will be valid.

The unregister mode unregisters Mirroring Controller as a Windows service.

L

If Mirroring Controller has not been started on the server that executes the command, commands for any mode other than the start mode, and status mode.

Execute this command as an instance administrator.

W

If Mirroring Controller has not been started on the server that executes the command, commands for any mode other than the start mode, status mode, register mode, and unregister mode terminate with an error.

Execute this command as a user with the "Administrator" privilege (operating system user ID that belongs to the Administrator group).

Until you start Mirroring Controller of standby server after starting Mirroring Controller of the primary server, operation can be started with only the primary server. Standby server is incorporated when you start the Mirroring Controller of standby server, and you should be able to operate in the multiplexing configuration.

Options

-a

Specify this option to stop Mirroring Controller on all servers.

-e

Specify this option to forcibly stop Mirroring Controller on the active server.

-f

Specify this option to enable automatic switching and disconnection of Mirroring Controller immediately after startup.

L

This option is the default of start mode.

W

This option is the default of start mode and register mode.

-F

Specify this option to disable automatic switching and disconnection immediately after startup of Mirroring Controller.

--async-connect-arbiter

Specify this option to connect the Mirroring Controller start process asynchronously to the Mirroring Controller arbitration process. This option can be specified to forcibly start Mirroring Controller if the Mirroring Controller arbitration process is not started.

Specify this option if using the Mirroring Controller arbitration server.

--arbiter

Specify this option to display the connection status of the Mirroring Controller arbitration process. This option can be specified if using status mode.

Specify this option if using the Mirroring Controller arbitration server.

--local-server *server_id*

If you run a simulation build of the primary and standby servers in a single server (for system testing, for example), specify this option to identify the server to be operated.

For *server_id*, specify the server identifier specified in the network.conf file. ASCII characters other than single-byte space can be specified in the server identifier. The operations will be executed as if the user has logged in to *server_id*.

--mc-only

Specify this option to start and stop only Mirroring Controller processes. At the start mode, this option can be specified only while the database instance is running. If this option is omitted, the database instance is simultaneously started and stopped.

-M *mcdir*

Specify the Mirroring Controller management directory.

ASCII characters can be specified in the directory path.

If this option is omitted, the value of the environment variable MCCONTROLDIR is used.

--force

Switching with this option specified can only be specified on the standby server. This option is used to perform switching forcibly after performing fencing on the primary server if communication with the Mirroring Controller process of the other server is not possible (due to a network issue between database servers or unresponsive server, for example), thus preventing normal switching. However, if the --no-fencing option is specified, fencing will not be performed on the primary server.

--no-fencing

When switching or disconnection is executed with the --force option specified, fencing of the server to be degraded is circumvented. Therefore, it is necessary for the user to isolate the server to be degraded from the cluster system in advance.

-P *password*

For the register mode, specify the password for the user who executed the command.

-S a[uto] | d[emand]

Specify the start type for the Windows service to be registered. You can choose auto or demand as the start type by specifying the entire word or just its first letter. The default is auto.

-w

Waits for operations to finish.

This option is the default of start mode.

This option is the default of start mode and register mode.

-W

Does not wait for operations to finish.

Environment variable

MCCONTROLDIR

Specifies the Mirroring Controller management directory.

ASCII characters can be specified in the directory path.

You can specify the -M option to override this value.

Diagnostics

0: Normal end

Other: Abnormal end

Notes

The message under execution might be output though the mc_ctl command is not being executed and, besides, it terminate abnormally when the server is downed while processing execution of this command, an automatic switch, and an automatic separation, and the communication between a primary server and the standby server is cut off. Besides, restart Mirroring Controller to solve this problem after confirming that the mc_ctl command is not in progress. Afterwards, execute a necessary operation.

If a time-out error occurs when the mc_ctl command is in progress, the messages may be different from the processes. Take the actions described in the "Action" section of the message.

Automatic switching and disconnection by the enable-failover mode, the disable-failover mode, and the start mode can be enabled/disabled only while Mirroring Controller is running. Therefore, if you want to enable/disable automatic switching and disconnection on starting, specify the -f option or -F option each time you start Mirroring Controller.

W

Use the start mode and stop mode to start and stop the Windows service. To do this, use the register mode to register Mirroring Controller as a Windows service in advance.

In case of postgresql.conf has any incorrect parameter when this command is executed, this command may be abnormally terminated. If this is the case, re-execute it again after correct the parameter in postgresql.conf.

If the arbitration server of Mirroring Controller is used, connection with the arbitration server will be performed on startup even if startup using start mode is executed with the -F option specified, or if executed with --local-server specified.

W

The mc_ctl command must be executed from "Administrator: Command Prompt". Right-click [Command Prompt], and then select [Run as administrator] from the menu to display the [Administrator: Command Prompt] window.

For operation using an arbitration server, the amount of time spent attempting to connect with the arbitration server is calculated, so the Mirroring Controller startup time may take longer.

Example

L

To start Mirroring Controller:

```
$ mc_ctl start -M /mcdir/inst1
```

W

To start Mirroring Controller:

```
> mc_ctl start -M D:\mcdir\inst1
```

Display details of mc_ctl status

```
mirroring status
----------------
(1)

server_id  host_role        host          host_status  db_proc_status       disk_status
(2)        (3)               (4)             (5)          (6)                    (7)
```

```
(1) Multiplexing status
    switchable        : Switchable
    switching         : Switching
    switched        : Switched (displayed when switching has finished and the degrading
operations status has been enabled)
    not-switchable    : Not switchable (displayed when a server is disconnected and
switching is not possible)
    unknown           : Unknown (*1)
    failover-disabled : Failover is disabled
(2) Server identifier
(3) Server role
    primary : Primary
    standby : Standby
    none(inactivated primary): No role
                              (primary is stopped or being defined as primary)
    none(inactivated standby): No role
                               (standby is stopped or being defined as primary)
(4) Host name or IP address
(5) Live/dead state of the server
    normal   : Normal operation
    abnormal : Abnormal
    unknown  : Unknown (*1)
(6) DBMS process status
    normal                         : Normal
    abnormal (abnormal process name (*2)) : Abnormal
    unknown                        : Unknown (*1)
(7) Disk status
    normal                             : Normal
    abnormal (abnormal disk type (*3)) : Abnormal
    unknown                            : Unknown (*1)

*1: Displayed when Mirroring Controller is stop state, the management network is
abnormal, or Mirroring Controller has failed or is unresponsive.

*2: The names of the DBMS processes in which the abnormality was detected are output. The
name has the following meaning: However, if multiple DBMS process issues are detected,
only the DBMS for which the first issue was detected is displayed.
     -postmaster: Process (postmaster) that accepts application connections
     -wal_sender or wal_receiver: Process (WAL sender or WAL receiver) that sends and
receives transaction logs
*3: The types of disks where the abnormality was detected are output separated by a
comma. The type has the following meaning:
     -data: Data storage disk
     -tran_log: Transaction log storage disk
     -tablespace: Tablespace storage disk
```

Display details of mc_ctl status (with the --arbiter option specified)

```
arbiter_id   host            status
(1)          (2)              (3)
```

```
(1) Arbitration server identifier
(2) Host name or IP address
(3) Mirroring Controller arbitration process connection status
    online  : Connected
    offline : Disconnected (*1)

*1: When Mirroring Controller is stopped, connections to the Mirroring Controller
arbitration process cannot be established, so it will be displayed as "offline".
```

Chapter 5 Connection Manager Commands

This chapter explains the Connection Manager commands.

5.1 cm_ctl

Name

L

cm_ctl - Start, stop or display the status of the conmgr process

W

cm_ctl - Start, stop, and display the status of the conmgr process, or register and unregister it as a Windows service

Synopsis

```
cm_ctl start [-D diretory] [-W] [--complete] [-t seconds]

cm_ctl stop [-D diretory] [-W] [-m {smart | fast | immediate}] [-t seconds]

cm_ctl status [-D directory] [-t seconds] [-i {all | instance | application}]
```

W

```
cm_ctl register [-D directory] [-N servicename] [-U username] [-P password]
                [-S a[uto] | d[emand] ] [-e source] [-W] [--complete] [-t seconds]

cm_ctl unregister [-N servicename]
```

Description

The start mode starts the conmgr process. The command returns at least after the heartbeat monitoring connection is completed with the primary server's instance.

When --complete is specified, wait until all instances configured in conmgr.conf have completed their heartbeat monitoring connections. You can set a timeout for these waits. The default of timeout is 60 seconds. Can be changed using the -t option. If it times out, it simply gives up waiting and the conmgr process remains up.

Note

If the primary is not among the instances configured in the cmgr.conf, use the -W option when starting Connection Manager with the cm_ctl command. Without the -W option, the cm_ctl command will not return until the primary connection is complete.

The stop mode sends a signal to the conmgr process to shut down and wait until the process disappeares.

The default of wait time is 60 seconds. Can be changed using the -t option. If it times out, it simply gives up waiting. There are three shutdown methods, "smart", "fast", and "immediate", specify with the -m option. The "smart" waits until all applications using the conmgr process run out of SQL connections before shutting down. The "fast" forces all applications using the conmgr process to disconnect from the conmgr process before shutting it down. As a result, the SQL connection for the application receives an error. The "immediate" terminates the conmgr process immediately. If nothing is specified, it stops in fast mode.

The status mode, if the commgr process exists, queries the conmgr process for instance and application information known to the conmgr process, and display them to standard output along with the state of conmgr itself.

The -i option allows you to specify what information to query. The "instance" queries information about the instance; The "application" queries information about the application; The "all" queries information about both. conmgr's own information is always displayed. The default time to wait for a query to return is 60 seconds. Can be changed using the -t option.

The register mode registers the conmgr process as a system service on Windows. The -S option allows you to select the service start type, either "auto" (to start the service automatically when the system starts) or "demand" (to start the service upon request).

The unregister mode unregisters the conmgr process from the Windows service.

Options

--complete

Wait until all instances configured in conmgr.conf have completed their heartbeat monitoring connections. If the When used with the -W option, the -W option takes precedence.

-D

-- directory=*directory*

Specify the directory where conmgr.conf is located. If omitted, it refers to the directory specified by the environment variable CMDATA. You cannot omit both.

-i {all | instance | application}

Specify the information to display the status.

-m

--mode={smart | fast | immediate}

Specify the mode of shutdown. The default is fast.

-t *seconds*

--timeout=*seconds*

Specify how long to wait for the operation to complete. The unit is seconds. The default is 60 seconds.

-W

--no-wait

In start mode, cm_ctl command returns immediately after forking conmgr process. In stop mode, the cm_ctl command returns without waiting for the process to disappeare.

-e *source*

The name of the cm_ctl event source to use for output to the event log when running as a Windows service. The default is conmgr. Note that this only controls messages sent from cm_ctl itself. After the conmgr process starts, the event source specified in event_source in conmgr.conf is used. If a failure occurs very early in the conmgr process startup, you may want to use the default event source name conmgr before the event source specified in conmgr.conf is set.

-N *servicename*

The name of the Windows service to register. This name is used as both the system name and the display name. The default is conmgr.

-P *password*

The password for the user who starts the service.

-S *start-type*

Start type of the Windows service to register. The start-type can be auto, demand, or one of the first characters of these two words. If you omit this option, auto is the default.

-U *username*

Username of the user who starts the service. Use the DOMAIN\username format for domain users.

Diagnostics

start mode or stop mode

0: Normal end

2: Timeout occurred

3: Unable to access the specified directory

Other than the above: None of the above

status mode

0: Normal end

3: Unable to access the specified directory

4: conmgr process does not exist

Other than the above: None of the above

register mode

0: Normal end

Other than 0: Abnormal end

unregister mode

0: Normal end

Other than 0: Abnormal end

Privileges

The conmgr process cannot be started by the administrator(e.g. superuser(root) on Linux, users with Windows administrator privileges).

Example

Display details of start mode

The block of information that can be specified with the -i option is used as a unit. There is one blank line between the blocks and no blank lines within the blocks. It includes one or more spaces between columns and between data.

```
$ cm_ctl status -i all
conmgr_status:
status pid
(1)     (2)
ready  3456

instance_information:
addr          port  database_attr
(3)           (4)   (5)
192.0.2.100   27500 standby
192.0.2.110   27500 primary
192.0.2.120   27500 standby
192.0.2.130   27500 unknown

application_information:
addr          port  pid   connected_time
(6)           (7)   (8)   (9)
127.0.0.1     12345 5678  2022-02-15 02:03:04
```

```
(1) Status of the conmgr process
    starting : The process is starting its startup sequence but is not ready to accept
               connections from clients.
    ready    : Ready to accept connections from clients.
```

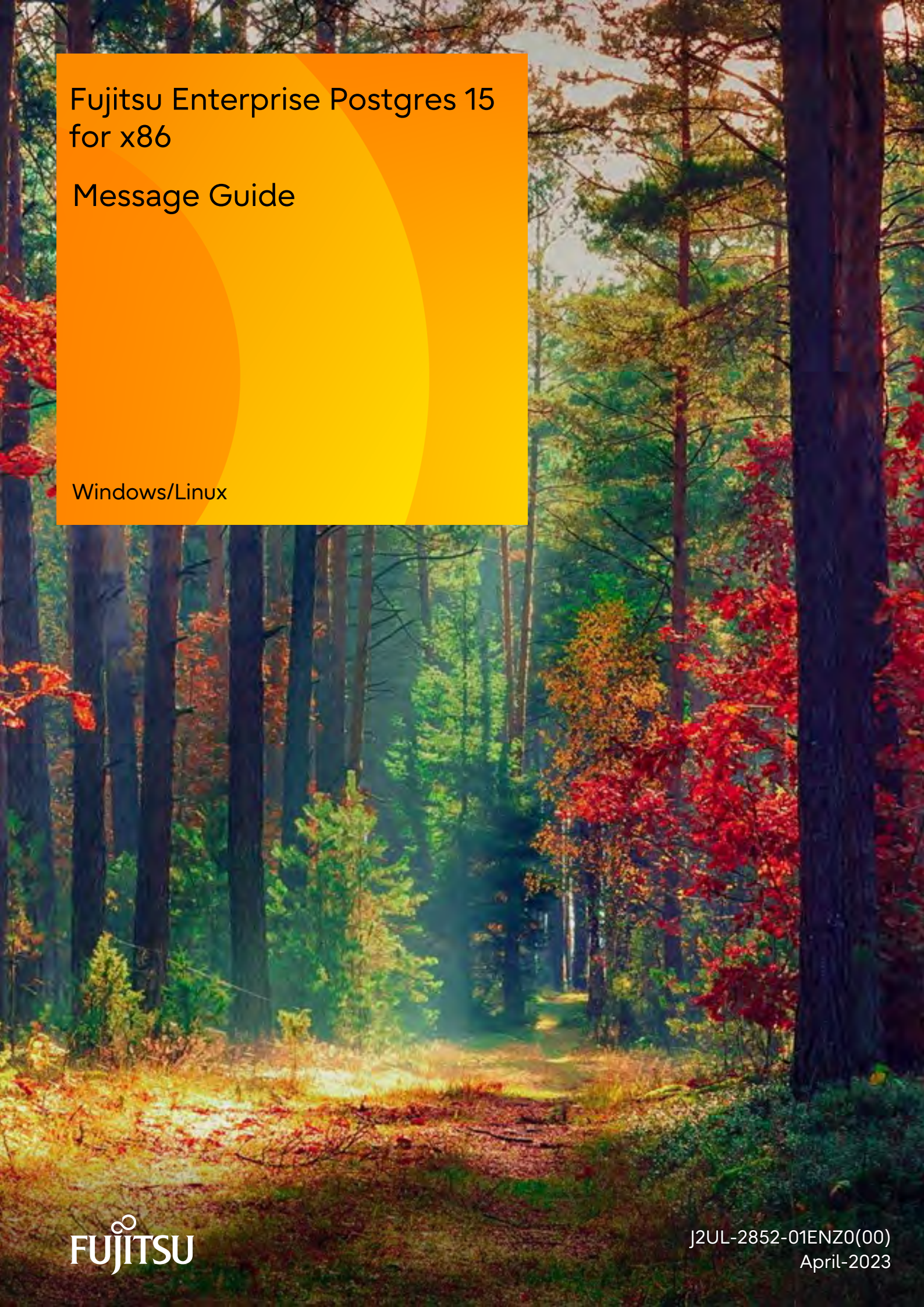
```
    stopping : It has received an end instruction and is starting the stop sequence.
    inactive : The conmgr process does not exist.
(2) PID of the conmgr process
(3) Host name or IP address of the instance
(4) Port number the postmaster listens on
(5) Status of the instance (primary | primary(read-only) | standby | unknown)
    primary            : Primary server
    primary(read-only) : Primary server (default transaction mode is read-only)
    standby            : Standby server
    unknown            : Unknown (*1)
(6) Connection source IP address for conmgr process
(7) Connection source (ephemeral) port number for conmgr process
(8) PID of the connection source
(9) Date and time conmgr process connection
    The ISO 8601 compliant date is followed by a blank, followed by the ISO 8601
    compliant second precision time.
    This representation is a PostgreSQL string representation of type timestamp.

*1) Displays when you cannot connect to the instance.
```

Fujitsu Enterprise Postgres 15 for x86

Message Guide

Windows/Linux



FUJITSU

J2UL-2852-01ENZ0(00)
April-2023

Preface

Purpose of this document

This document explains the messages output by Fujitsu Enterprise Postgres.

Intended readers

This document is intended for the following readers:

- Persons using Fujitsu Enterprise Postgres

Prerequisites

Knowledge of the following topics is required to read this document:

- A general understanding of computers
- Jobs
- PostgreSQL
- Linux
- Windows

Structure of this document

This document is structured as follows:

Chapter 1 Overview of Messages

This chapter explains the format in which messages are output.

Chapter 2 Message List

This chapter explains the output messages.

Chapter 3 Mirroring Controller Messages

This chapter explains messages output by Mirroring Controller.

Export restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Issue date and version

Edition 1.0: April 2023

Copyright

Copyright 2015-2023 Fujitsu Limited

Chapter 1 Overview of Messages

This chapter explains the format of messages.

1.1 Message Format

This section explains the format of messages.

- Output format
- Components

1.1.1 Output Format

The message output format is as follows:

Format of messages returned to an application

```
msgType: msgText
```

Format of messages output to the server message log

```
SQLSTATE: date [processID]: [internalCode-1] user = userName,db = dbName,remote =
clientIpAddress(portNumber) app = appName msgType: msgText
```

See

Refer to "PostgreSQL Error Codes" under "Appendixes" in the PostgreSQL Documentation for information on SQLSTATE.

Note

- Notes on monitoring messages output to the server message log

  Use SQLSTATE to monitor server messages, noting the following:

  - Configuration method

    Refer to "Error Log Settings" under "Setup" in the Installation and Setup Guide for Server for details.

  - Note

    - The user name, client IP address (port number), and application name may sometimes be blank.

- Notes on monitoring messages returned to an application

  You can output SQLSTATE to a message to be returned to an application. The following explains how to configure the settings for outputting SQLSTATE and gives cautions to be observed when doing so.

  - Configuration methods

    - In the SET statement, set the log_error_verbosity parameter to VERBOSE.
    - For an application that uses the C language library, use the PQsetErrorVerbosity function to set message redundancy to PQERRORS_VERBOSE.

  - Notes

    - SQLSTATE is output only to messages to be returned to applications that use the C language library.
    - In some cases, *userName*, *clientIpAddress*(*portNumber*), and *applicationName* may be blank.

- If the email address "pgsql-bugs@postgresql.org" is output to the message and the cause of the error cannot be identified, contact Fujitsu technical support.

Example

Message output to the server message log

```
3D000: 2017-07-10 19:41:05 JST [13899]: [1-1] user = fepuser,db = fep,remote =
127.0.0.1(51902) app = [unknown] FATAL:  database "fep" does not exist
```

1.1.2 Components

This section explains the components of a message.

Message type

The message type indicates the type of error denoted in the message.

The message type will be one of the following:

- Information (INFO, NOTICE, LOG, DEBUG)

  This message type denotes a notification from the system, not an error. There is no need to take action.

- Warning (WARNING)

  This message type denotes that no error occurred but confirmation or action is required by the user. Take the actions described in the "Action" section of the message.

- Error (ERROR, FATAL, PANIC)

  This message type denotes that an error has occurred. Take the actions described in the "Action" section of the message.

- Supplementary information (DETAIL, HINT, QUERY, CONTEXT, LOCATION, STATEMENT)

  This message type denotes supplementary information relating to the previous message. If the message was output in English, the message type will also be in English. (Detail, hint, query, context, location, statement)

Message text

The text of the message reports the status of the system or an error in the system.

The notation "@numeric character@" that appears in "Chapter 2 Message List" indicates an embedded character string. A character string is output to a message that is actually output.

If a message locale other than 'ja' is specified, messages added by Fujitsu Enterprise Postgres will be output in English.

For other message locales, the messages are output in English.

1.2 Mirroring Controller Message Format

This section explains the format of messages output by the Mirroring Controller.

Mirroring Controller messages are output to the following locations:

L - System log

W - Event log

Output format

```
programName[processId]: msgType: msgText (msgNumber)
```

For *programName*, use the value of the syslog_ident parameter or event_source parameter defined in the *serverIdentifier*.conf file.

L

The message types output by Mirroring Controller, their severity, and their corresponding value in the system log are shown in the table below.

L

Table 1.1 Message type, severity, and corresponding value in the system log

Message type	Severity	Meaning	System log
INFO	Information	Provides information not categorized as LOG or NOTICE.	INFO
LOG		Provides information recognized as a particularly important event in tracing the operation history. (Example: Automatic switch is complete)	
NOTICE	Notice	Outputs information that takes into account the user instructions within the program in response to an executed or automatically executed process.	NOTICE
WARNING	Warning	Provides a warning, for example it will soon be impossible to maintain multiplexing capabilities.	WARNING
ERROR	Error	Reports that an error other than FATAL or PANIC has occurred.	ERROR
FATAL		Reports that an error has been detected requiring system recovery in one of the multiplexed database systems. It also reports the content and cause of the error.	CRIT
PANIC		Reports that an error has been detected requiring immediate system recovery in all multiplexed database systems. It also reports the content and cause of the error.	ALERT

W

The message types output by Mirroring Controller, their severity, and their corresponding value in the event log are shown in the table below.

W

Table 1.2 Message type, severity, and corresponding value in the event log

Message type	Severity	Meaning	Event log
INFO	Information	Provides information not categorized as LOG or NOTICE.	INFORMATION
LOG		Provides information recognized as a particularly important event in tracing the operation history. (Example: Automatic switch is complete)	
NOTICE	Notice	Outputs information that takes into account the user instructions within the program in response to an executed or automatically executed process.	
WARNING	Warning	Provides a warning, for example it will soon be impossible to maintain multiplexing capabilities.	WARNING

Message type	Severity	Meaning	Event log
ERROR	Error	Reports that an error other than FATAL or PANIC has occurred.	ERROR
FATAL		Reports that an error has been detected requiring system recovery in one of the multiplexed database systems. It also reports the content and cause of the error.	
PANIC		Reports that an error has been detected requiring immediate system recovery in all multiplexed database systems. It also reports the content and cause of the error.	

The message severity has the following meanings:

- Information

  Informational status. A message that was reported by the system is displayed. No action is required.

- Notice

  Informational status, but a message that should be noted is displayed. If necessary, take the actions described in the "Action" section of the message.

- Warning

  No error has occurred, but the user is requested to check, and take action. Take the actions described in the "Action" section of the message.

- Error

  An error has occurred. Take the actions described in the "Action" section of the message.

Output destination server

Mirroring Controller messages are output by the database server. When using an arbitration server, Mirroring Controller messages are also output by the arbitration server.

- Messages with a message number that starts with "MCA" are output by the database server.
- Messages with a message number that starts with "MCR" are output by the arbitration server.

Chapter 2 Message List

This chapter explains the output messages.

For messages not described in this chapter, check the content of the message and take appropriate action. If the problem cannot be resolved, contact Fujitsu technical staff(support).

needs to set at least @1@ to "max_worker_processes"

[Description]

An error occurred during execution of the application or command.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the application is written correctly and the command is being used correctly.

@1@ total checksum verification failure

[Description]

By default, checksums are verified and checksum failures are reported for any possible page corruptions

[System Processing]

Continues processing.

[Action]

Check the message text and confirm that the event indicated in supplementary information reported by the system is a planned event.

@1@ and @2@ are incompatible options

[Description]

The input parameter in invalid.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the application is written correctly and the command is being used correctly.

@1@ can only be called in a sql_drop event trigger function

[Description]

can only be called in a sql_drop event trigger function

[System Processing]

Processing will be aborted.

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ can only be called in a table_rewrite event trigger function

[Description]

The function pg_event_trigger_table_rewrite_oid() can only be called in a table_rewrite event trigger function

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ can only be called in an event trigger function

[Description]

The function pg_event_trigger_ddl_commands() can only be called in a event trigger function

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ cannot be applied to a function

[Description]

SQL row locking clause such as "FOR UPDATE" cannot be applied to a function

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ cannot be applied to a join

[Description]

SQL row locking clause such as "FOR UPDATE" cannot be applied to a join

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ cannot be applied to a WITH query

[Description]

SQL row locking clause such as "FOR UPDATE" cannot be applied to a WITH query

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ cannot be applied to the nullable side of an outer join

[Description]

SQL row locking clause cannot be applied to the nullable side of an outer join

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ cannot be applied to VALUES

[Description]

SQL row locking cannot be applied to VALUES

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ cannot be executed from VACUUM or ANALYZE

[Description]

commands VACUUM or ANALYZE cannot be executed from VACUUM or ANALYZE

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ cannot be used as a role name here

[Description]

Reserved name cannot be used as a role name here

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ cause an error in heartbeat: @2@

[Description]

Cause an error in heartbeat.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

@1@ causes an error at application connection: @2@

[Description]

Causes an error at application connection.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

@1@ function has @2@ columns available but @3@ columns specified

[Description]

An error occurred, when specified column of the function is different than the available column

[System Processing]

Continues processing.

[Action]

Investigate the cause of the error from the message body, and then remove the cause.

@1@ is not allowed with aggregate functions

[Description]

The return type of function LCS_asString is not allowed with aggregate functions.In this error case,the return type of LCS_asString is "FOR some"

[System Processing]

Processing will be aborted.

[Action]

To investigate the cause of the occurrence from the message, and remove cause

@1@ is not allowed with DISTINCT clause

[Description]

SQL row locking clause such as "FOR UPDATE" is not allowed with DISTINCT clause

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ is not allowed with GROUP BY clause

[Description]

SQL row locking clause such as "FOR UPDATE" is not allowed with GROUP BY clause

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ is not allowed with HAVING clause

[Description]

SQL row locking clause such as "FOR UPDATE" is not allowed with HAVING clause

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ is not allowed with set-returning functions in the target list

[Description]

SQL row locking clause such as "FOR UPDATE" is not allowed with set-returning functions in the target list

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ is not allowed with UNION/INTERSECT/EXCEPT

[Description]

Locking clause is not allowed with UNION/INTERSECT/EXCEPT

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ is not allowed with window functions

[Description]

SQL row locking clause such as "FOR UPDATE" is not allowed with window functions

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ is not an ordered-set aggregate, so it cannot have WITHIN GROUP

[Description]

The function is not an ordered-set aggregate, so it cannot have WITHIN GROUP

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ is not VCI index

[Description]

An error occurred during execution of the application or command.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the application is written correctly and the command is being used correctly.

@1@ must be called in REPEATABLE READ isolation mode transaction

[Description]

When creating a new replication slot, we must put CREATE_REPLICATION_SLOT and USE_SNAPSHOT in REPEATABLE READ isolation mode transAction.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the application is written correctly and the command is being used correctly.

@1@ must be called inside a transaction

[Description]

When creating a new replication slot, we must put CREATE_REPLICATION_SLOT and USE_SNAPSHOT in a transAction.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the application is written correctly and the command is being used correctly.

@1@ must not be called in a subtransaction

[Description]

When creating a new replication slot, we must not put CREATE_REPLICATION_SLOT and USE_SNAPSHOT in a subtransAction.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the application is written correctly and the command is being used correctly.

@1@ must not be called inside a transaction

[Description]

When creating a new replication slot, it is not allowed to put CREATE_REPLICATION_SLOT and EXPORT_SNAPSHOT in a transAction.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the application is written correctly and the command is being used correctly.

@1@ must specify unqualified relation names

[Description]

SQL row locking clause such as "FOR UPDATE" must specify unqualified relation names

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ needs a slot to be specified using --slot

[Description]

needs a slot to be specified using --slot

[System Processing]

Processing aborts

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

@1@ the prepared transaction with gid: "@2@"

[Description]

The prepared transaction has been completed.

[System Processing]

Continues processing.

[Action]

No action required.

@1@(@2@) failed for @3@ address "@4@": @5@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

@1@: Copy of objects for allnodes and Setup of replication table weren't executed

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: could not connect to database "@2@"

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: could not connect to database "postgres" or "template1"

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: could not create file "@2@" for writing

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: could not locate my own executable path

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: could not open file "@2@" for writing

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: could not remove initialize.signal file

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: could not remove scaleout.signal file

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: could not remove scaleout_passfile

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: Failed connect to coodinator

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: Failed copy replication table

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: Failed define objects to datanode

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: Failed make replication object

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: Failed open initialize.signal

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: Failed open postmaster.pid

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: Failed read initialize.signal

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: initialize.signal does not exist

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: out of memory

[Description]

There was insufficient free space in the server's memory during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Estimate memory usage and take the following action:

- If the number of simultaneous connections from client applications is high, reduce it.

- If the number of simultaneous SQL executions is high, reduce it.

@1@: pg_dumpall failed

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: postmaster.pid does not exist

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: query failed: @2@

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: query was: @2@

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

@1@: Stop sever process and re-execute pg_ctl start with longer timeout

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

"@1@" is not a directory

[Description]

Not a directory.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

"@1@" is not pid file: number of line is not enough(expect 4 but @2@)

[Description]

Not pid file: number of line is not enough.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

"root" execution of the conmgr is not permitted.The conmgr must be started under an unprivileged user ID to preventpossible system security compromise. See the documentation formore information on how to properly start the conmgr.

[Description]

"root" execution of the conmgr is not permitted. The conmgr must be started under an unprivileged user ID to prevent possible system security compromise. See the documentation for more information on how to properly start the conmgr.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

"VARYING" cannot have a level number greater than 48

[Description]

Variables that have greather than 48 level have VARYING clause.

[System Processing]

Precompling will be aborted.

[Action]

Remove "VARYNING" clause or change the level number.

Is the server running locally and accepting connections on that socket?

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

Is the server running on that host and accepting TCP/IP connections?

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

a new master encryption key has been set (@1@/@2@)

[Description]

A new encryption key has been set. The current number of save keys is the number on the left, and the maximum number of save keys is the number on the right.

[System Processing]

This is an information messaage when TDE_h is loaded.

[Action]

Provides notification of the current number of saved keys. If the maximum number is reached, the oldest key is deleted first. Check that the operation is as expected, and if it is not as expected, consider modifying postgresql.conf.

aborting any active transactions

[Description]

Rollback any active transactions because the database system is being requested to shut down.

[System Processing]

Continues processing.

[Action]

Retry any necessary applications or commands after restarting the database system.

All directory entries in pg_tblspc/ should be symbolic links.

[Description]

An error occurred due to unexpected directory entry found at specific path

[System Processing]

Cancels processing.

[Action]

Remove those unexpected directories, or set allow_in_place_tablespaces to ON transiently to let recovery complete.

---

another conmgr process is running

[Description]

Another conmgr process is running.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

---

another conmgr process is shutting down

[Description]

Another conmgr process is shutting down.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

---

another conmgr process is starting up

[Description]

Another conmgr process is starting up.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

---

another conmgr process(PID @1@) is running

[Description]

Another conmgr process is running.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

---

application connection is closed by remote in @1@

[Description]

Application connection is closed by remote.

[System Processing]

Processing continues.

[Action]

Check log of remote system.

---

array slice subscript must provide both boundaries

[Description]

An error occurred during execution of the application or command.

[System Processing]

Processing will be aborted.

[Action]

When assigning to a slice of an empty array value, slice boundaries must be fully specified.

---

at least one COORDINATOR is needed

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

---

authentication identifier set more than once

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

bind message has @1@ parameter formats but @2@ parameters

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

bind message has @1@ result formats but query has @2@ columns

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

bind message supplies @1@ parameters, but prepared statement "@2@" requires @3@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

canceling authentication due to timeout

[Description]

Timeout occurred during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Check the following:

- If executing SQL that outputs a large volume of search results, add a conditional expression to filter the results further.

- If numerous SQLs are being simultaneously executed, reduce the number of simultaneously executed SQLs.

- If a large volume of data is to be updated in a single transaction, modify the SQL to reduce the volume of data to be updated in a single transaction.

- If executing a complex SQL, modify it to a simple SQL.

- Check if there are any problems in the network.

- Before conducting maintenance that involves the processing of a large volume of data, use the SET statement to temporarily increase the value of maintenance_work_mem.

canceling statement due to conflict with recovery

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

canceling statement due to statement timeout

[Description]

Timeout occurred during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Check the following:

- If executing SQL that outputs a large volume of search results, add a conditional expression to filter the results further.

- If numerous SQLs are being simultaneously executed, reduce the number of simultaneously executed SQLs.

- If a large volume of data is to be updated in a single transaction, modify the SQL to reduce the volume of data to be updated in a single transaction.

- If executing a complex SQL, modify it to a simple SQL.

- Check if there are any problems in the network.

- Before conducting maintenance that involves the processing of a large volume of data, use the SET statement to temporarily increase the value of maintenance_work_mem.

---

cannot @1@ "@2@" because it has pending trigger events

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

cannot @1@ "@2@" because it is being used by active queries in this session

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

cannot change client_encoding during a parallel operation

[Description]

This error occurs if a function wants to set client_encoding and is invoked by the parallel query.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that client_encoding is not set by the parallel query operation.

---

cannot create temporary tables during recovery

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

cannot enter pipeline mode, connection not idle

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

cannot exit pipeline mode while busy

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

cannot exit pipeline mode with uncollected results

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

cannot replace existing key

[Description]

An error occurred during execution of the application or command.

[System Processing]

Processing will be aborted.

[Action]

Try using the function jsonb_set to replace the key value.

cannot uniquely identify the id

[Description]

Could not uniquely identify the id.

[System Processing]

Processing will be aborted.

[Action]

Please refer to the server log, and determine the cause of the error.

certificate does not match private key file "@1@": @2@

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

checkpoints are occurring too frequently (@1@ second apart)

[Description]

Checkpoints are occurring too frequently.

[System Processing]

Continues processing.

[Action]

Consider increasing the value of configuration parameter max_wal_size.

closing all application connections

[Description]

Closing all application connections.

[System Processing]

Processing continues.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

cmdata directory "@1@" does not exist

[Description]

Cmdata directory does not exist.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

---

collect shardinfo connection failed: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

conmgr process is ready

[Description]

Conmgr process is ready.

[System Processing]

Continues processing.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

---

conmgr process is starting

[Description]

Conmgr process is starting.

[System Processing]

Processing continues.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

---

conmgr process shutting down

[Description]

Conmgr process shutting down.

[System Processing]

Continues processing.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

---

conmgr process stopped

[Description]

Conmgr process stopped.

[System Processing]

Processing continues.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

connection manager directory "@1@" does not exist

[Description]

Connection manager directory does not exist.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

connection to client lost

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

connection to server at "@1@" (@2@), port @3@ failed:

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

connection to server at "@1@", port @2@ failed:

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

connection to server on socket "@1@" failed:

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

COPY statement cannot be null

[Description]

COPY statement cannot be null.

[System Processing]

Processing will be aborted.

[Action]

Please specify COPY FROM statement.

could not accept application connection: @1@

[Description]

Could not accept application connection.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

could not accept heartbeat connection: @1@

[Description]

Could not accept heartbeat connection.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

could not accept new connection: @1@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

could not accept SSL connection: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

could not accept SSL connection: EOF detected

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

could not access to cmdata directory "@1@": @2@

[Description]

Could not access to cmdata directory.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not access to connection manager directory "@1@": @2@

[Description]

Could not access to connection manager directory.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not add node: host=@1@ hostaddr=@2@ port=@3@

[Description]

Could not add node.

[System Processing]

Processing aborts.

[Action]

Check messages near this message.

---

could not be ready for next command: @1@

[Description]

Could not be ready for next command.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not bind @1@ address "@2@": @3@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

---

could not bind port @1@ with following causes

[Description]

Could not bind port with following causes.

[System Processing]

Processing aborts.

[Action]

Check messages near this message.

---

could not bind to local address "@1@": @2@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

---

could not change directory to "@1@": @2@

[Description]

Could not change directory.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not check the existence of the backend with PID @1@: @2@

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

could not connect due to the following errors

[Description]

Could not connect due to the following errors.

[System Processing]

Processing aborts.

[Action]

Check messages near this message.

---

could not connect to conmgr process: @1@

[Description]

Could not connect to conmgr process.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not connect to database @1@: out of memory

[Description]

There was insufficient free space in the server's memory during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Estimate memory usage and take the following action:

- If the number of simultaneous connections from client applications is high, reduce it.

- If the number of simultaneous SQL executions is high, reduce it.

---

could not connect to Ident server at address "@1@", port @2@: @3@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

---

could not connect to publisher when attempting to drop replication slot "@1@": @2@

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

could not connect to the primary server: @1@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

---

could not create @1@ socket for address "@2@": @3@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

---

could not create an additional interval event: @1@

[Description]

Could not create an additional interval event.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not create archive status file "@1@": @2@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not create bufferevent: @1@

[Description]

Could not create bufferevent.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not create conmgr process: @1@

[Description]

Could not create conmgr process.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not create directory "@1@": @2@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not create file "@1@": @2@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not create lock file "@1@": @2@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not create missing directory "@1@": @2@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not create pid file "@1@": @2@

[Description]

Could not create pid file.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not create pipe for syslog: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

could not create shared memory segment: @1@

[Description]

There was insufficient free space in the database server's shared memory during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Estimate memory usage and take the following action:

- If the number of simultaneous connections from client applications is high, reduce it.

- If the number of simultaneous SQL executions is high, reduce it.

---

could not create ShmemIndex entry for data structure "@1@"

[Description]

There was insufficient free space in the database server's shared memory during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Estimate memory usage and take the following action:

- If the number of simultaneous connections from client applications is high, reduce it.

- If the number of simultaneous SQL executions is high, reduce it.

---

could not create socket for Ident connection: @1@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

---

could not create socket: @1@

[Description]

There was insufficient free space in the server's memory during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Estimate memory usage and take the following action:

- If the number of simultaneous connections from client applications is high, reduce it.

- If the number of simultaneous SQL executions is high, reduce it.

---

could not drop relation mapping for subscription "@1@"

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

could not duplicate socket @1@ for use in backend: error code @2@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

---

could not establish heartbeat connection

[Description]

Could not establish heartbeat connection.

[System Processing]

Processing aborts.

[Action]

Check messages near this message.

---

could not establish SSL connection: @1@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

---

could not extend file "@1@": @2@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not extend file "@1@": wrote only @2@ of @3@ bytes at block @4@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not get a result from conmgr process: @1@

[Description]

Could not get a result from conmgr process.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

could not get function(@1@) from library "@2@" :@3@

[Description]

Failed to load a function in the PKCS11 library.

[System Processing]

Processing will be aborted.

[Action]

Check that the PKCS11 library version and/or path are correct and take action.

could not get home directory to locate root certificate fileEither provide the file or change sslmode to disable server certificate verification.

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

could not get peer credentials: @1@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

could not get string format address: @1@

[Description]

Could not get string format address.

[System Processing]

Processing continues.

[Action]

Check log of Operating System.

---

could not initialize checksum of file "@1@"

[Description]

There was insufficient free space in the database server's shared memory during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Estimate memory usage and take the following action:

- If the number of simultaneous connections from client applications is high, reduce it.

- If the number of simultaneous SQL executions is high, reduce it.

---

could not initialize SSL connection: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

could not interpret result from server: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

could not listen for socket: @1@

[Description]

Could not listen for socket.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

could not listen on @1@ address "@2@": @3@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

could not load locale "@1@"

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

could not load private key file "@1@": @2@

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

could not lock semaphore: error code @1@

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

could not notify application: @1@

[Description]

Could not notify application.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not obtain lock on row in relation "@1@"

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

could not open certificate file "@1@": @2@

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

---

could not open file "@1@": @2@

[Description]

Could not open file.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

could not open pid file "@1@": @2@

[Description]

Could not open pid file.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

could not parse connection string: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

could not prepare for heartbeat: @1@

[Description]

Could not prepare for heartbeat.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

could not prepare statement to fetch file contents: @1@

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

could not read certificate file "@1@": @2@

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

---

could not read from logger pipe: @1@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

---

could not read from pid file "@1@": @2@

[Description]

Could not read from pid file.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not read lock file "@1@": @2@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

could not read root certificate file "@1@": @2@

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

could not read symbolic link "@1@": @2@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

could not receive a heartbeat packet: @1@

[Description]

Could not receive a heartbeat packet.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

could not receive data from client: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

could not receive data from server: @1@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

---

could not receive database system identifier and timeline ID from the primary server: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

could not receive response from Ident server at address "@1@", port @2@: @3@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

---

could not remove pid file "@1@": @2@

[Description]

Could not remove pid file.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not reply all application info: @1@

[Description]

Could not reply all application info.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not reply all node info: @1@

[Description]

Could not reply all node info.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not reply for startup: @1@

[Description]

Could not reply for startup.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not reply target node info: @1@

[Description]

Could not reply target node info.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not resolve address: @1@

[Description]

Could not resolve address.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

Could not resolve client IP address to a host name: @1@.

[Description]

Supplementary information was output.

[System Processing]

None.

[Action]

Refer to this message together with the message that was output immediately beforehand.

---

could not seek pid file "@1@": @2@

[Description]

Could not seek pid file.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not send @1@: @2@

[Description]

Could not send.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not send data to client: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

could not send data to server: @1@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

---

could not send query to Ident server at address "@1@", port @2@: @3@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

---

could not send signal to conmgr process: @1@

[Description]

Could not send signal to conmgr process

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not send signal(@1@) to conmgr process(PID: @2@): @3@

[Description]

Could not send signal to conmgr process.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not set socket to nonblocking mode: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

could not set SSL Server Name Indication (SNI): @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

could not set SSL socket: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

could not start conmgr process due to setsid() failure: @1@

[Description]

Could not start conmgr process due to setsid() failure.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not start conmgr process in time

[Description]

Could not start conmgr process in time.

[System Processing]

Processing aborts.

[Action]

Check log of conmgr process.

---

could not start conmgr process: @1@

[Description]

Could not start conmgr process.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not start conmgr process

[Description]

Could not start conmgr process.

[System Processing]

Processing aborts.

[Action]

Check log of conmgr process.

---

could not stat file "@1@": @2@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not stop conmgr process in time

[Description]

Could not stop conmgr process in time.

[System Processing]

Processing aborts.

[Action]

Check log of conmgr process.

---

could not sync to pid file "@1@": @2@

[Description]

Could not sync to pid file.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

Could not translate client host name "@1@" to IP address: @2@.

[Description]

Supplementary information was output.

[System Processing]

None.

[Action]

Refer to this message together with the message that was output immediately beforehand.

---

could not translate name

[Description]

An error occurred while translating domain name to Kerberos realm name in SSL.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the domain name is written correctly.

---

could not try-lock semaphore: error code @1@

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

could not write archive status file "@1@": @2@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not write block @1@ in file "@2@": @3@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not write block @1@ in file "@2@": wrote @3@ of @4@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not write block @1@ in file "@2@": wrote only @3@ of @4@ bytes

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not write bootstrap write-ahead log file: @1@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not write file "@1@": @2@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not write pid file "@1@": @2@

[Description]

Could not write pid file.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

could not write server file "@1@": @2@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not write temporary statistics file "@1@": @2@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not write to COPY file: @1@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not write to file "@1@": @2@

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

---

could not write to log file @1@ at offset @2@, length @3@: @4@

[Description]

There are the following cases:

- failed to write transaction log file

- failed to write transaction log file on backup data storage destination

[System Processing]

Processing will be aborted.

[Action]

Lack of storage space or malfunction of storage allocating the file shown in this message is considered.

If it's ture, recover the database system accoring to "Actions when an Error Occurs" of "Operation Guide" or "Cluster Operation Guide (Database Multiplexing)".

If it's not true, identify the cause according to the informations in this message such as errno, and work around.

The following major causes are considered.

- the file has no permission or the permission has been changed

- power of the storage allocating the file has been turned off

- unmounted the storage allocating the file

- another process or human operated the file

- the storage allocating the file has crashed

---

could not write to pid file "@1@": @2@

[Description]

Could not write to pid file.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

database is not accepting commands to avoid wraparound data loss in database "@1@"

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

database is not accepting commands to avoid wraparound data loss in database with OID @1@

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

deadlock detected

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

disconnection: session time: @1@:@2@:@3@.@4@ user=@5@ database=@6@ host=@7@@8@@9@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

either backend_host or backend_hostaddr is needed for instance #@1@

[Description]

Either backend_host or backend_hostaddr is needed for instance.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

---

ending log output to stderr

[Description]

Ending log output to stderr.

[System Processing]

Continues processing.

[Action]

Future log output will be output to the log destination.

---

error is occured for completion of asynchronous connect for heartbeat: @1@

[Description]

Error is occured for completion of asynchronous connect for heartbeat.

[System Processing]

Processing aborts.

[Action]

Check log of Operating System.

---

error sending command to database "@1@": @2@

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

Execution of conmgr by a user with administrative permissions is notpermitted.The conmgr must be started under an unprivileged user ID to preventpossible system security compromises. See the documentation formore information on how to properly start the conmgr.

[Description]

Execution of conmgr by a user with administrative permissions is not permitted. The conmgr must be started under an unprivileged user ID to prevent possible system security compromises. See the documentation for more information on how to properly start the conmgr.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

---

expected authentication request from server, but received @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

expected GSS response, got message type @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

expected password response, got message type @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

expected SSPI response, got message type @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

failed to add new prepared transaction "@1@" by worker(@2@)

[Description]

Failed to add new prepared transaction.

[System Processing]

Processing will be aborted.

[Action]

Please refer to the server log, and determine the cause of the error.

---

failed to delete the record with id: @1@(SPI returned: @2@)

[Description]

Could not delete the row in pgx_loader_state table.

[System Processing]

Processing will be aborted.

[Action]

Please refer to the server log, and determine the cause of the error.

---

failed to execute SPI returned @1@

[Description]

Could not execute SPI.

[System Processing]

Processing will be aborted.

[Action]

Please refer to the server log, and determine the cause of the error.

---

failed to fetch id column from "@1@" by worker(@2@)

[Description]

Failed to fetch the id column.

[System Processing]

Processing will be aborted.

[Action]

Please refer to the server log, and determine the cause of the error.

---

failed to send GSSAPI negotiation response: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

failed to update state column of record with id: @1@

[Description]

Failed to update the state column.

[System Processing]

Processing will be aborted.

[Action]

Please refer to the server log, and determine the cause of the error.

---

function call message contains @1@ argument formats but @2@ arguments

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

function call message contains @1@ arguments but function requires @2@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

heartbeat connect is refused: @1@

[Description]

Heartbeat connect is refused.

[System Processing]

Processing aborts.

[Action]

Check log of remote system.

heartbeat connect is timed out

[Description]

Heartbeat connect is timed out.

[System Processing]

Processing aborts.

[Action]

Check log of remote system.

heartbeat connection is established

[Description]

Heartbeat connection is established.

[System Processing]

Processing continues.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

heartbeat is timed out

[Description]

Heartbeat is timed out.

[System Processing]

Processing aborts.

[Action]

Check log of remote system.

host name must be specified for a verified SSL connection

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

host name must be specified

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

incomplete message from client

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

incomplete multibyte character

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

incomplete startup packet

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

Increase watchdog.max_hb_connections.

[Description]

Increase watchdog.max_hb_connections.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

internal error: received unexpected database pattern_id @1@

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

internal error: received unexpected relation pattern_id @1@

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

invalid argument size @1@ in function call message

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

invalid connection option "@1@"

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

invalid DESCRIBE message subtype @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

invalid frontend message type @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

invalid length of startup packet

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

invalid message format

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

invalid message length

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

invalid parameter '@1@' in configuration file "@2@" line @3@

[Description]

Invalid parameter in configuration file.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

invalid password packet size

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

invalid pid file "@1@"

[Description]

Invalid pid file.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

invalid record length at @1@/@2@: wanted @3@, got @4@

[Description]

invalid record length was found on archive log or transaction log data.

[System Processing]

The following causes could be considered.

- if the log level is information(INFO, NOTICE, LOG, DEBUG)

Continue processing.

- if the log level is error(ERROR, FATAL, PANIC)

Processing will be aborted.

[Action]

If the log level is information(INFO, NOTICE, LOG, DEBUG), no action is required.

When the log level is error(ERROR, FATAL, PANIC), take either of the following actions.

- if this message is output during starting instance

Please restore according to "Deal at the time of abnormality" of "Operation Guide" or "Cluster Operation Guide (Database Multiplexing)".

- if this message is output during recovering

Cannot continue to recover with the current backup data because an archive log in the backup has an error.

Recover from the other backup data.

---

invalid response from primary server

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

invalid socket: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

---

invalid standby message type "@1@"

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

invalid startup packet layout: expected terminator as last byte

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

invalid string in message

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

invalid timeout "@1@"

[Description]

Invalid timeout specified.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

invalid value for integer parameter in configuration file "@1@" line @2@

[Description]

Invalid value for integer parameter in configuration file.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

invalid value for parameter '@1@'

[Description]

Invalid value for parameter.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

invalid value for parameter '@1@' in configuration file "@2@" line @3@

[Description]

Invalid value for parameter in configuration file.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

invalid value for parameter '@1@' of instance #@2@

[Description]

Invalid value for parameter of instance.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

Keystore does not require passphrase.

[Description]

keystore-passphrase command line option and PGX_KEYSTORE_PASSPHRASE environment variable should not be set when using TDE_z extension.

[System Processing]

Processing will be aborted.

[Action]

Remove keystore-passphrase from command line option or unset PGX_KEYSTORE_PASSPHRASE environment variable before starting the server with TDE_z extension loaded.

lack of max_prepared_transactions

[Description]

Lack of max_prepared_transactions.

[System Processing]

Processing will be aborted.

[Action]

Please increase max_prepared_transactions.

level number 77 could not be used for group item

[Description]

A group data item contains the 77 level variable.

[System Processing]

Precompling will be aborted.

[Action]

Remove the variable from the group data item.

level number 77 could not be used in TYPEDEF statement

[Description]

77 level variable is used in the TYPEDEF statement.

[System Processing]

Precompling will be aborted.

[Action]

Change the level number.

Limits the total size of all temporary files used by each process.

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

line @1@ too long in service file "@2@"

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

listening on @1@ port @2@

[Description]

Listening on port.

[System Processing]

Processing continues.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

logical replication target relation "@1@.@2@" uses system columns in REPLICA IDENTITY index

[Description]

Logical replication target relation uses some of the system columns to identify rows which are updated or deleted.

[System Processing]

Processing will be aborted.

[Action]

Usage of system columns for REPLICA IDENTITY should be avoided as these data vary across databases.

---

lost synchronization with server: got message type "@1@", length @2@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

---

materialize mode required, but it is not allowed in this context

[Description]

Materialize mode required, but it is not allowed in this context.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

---

max_prepared_transactions must be set at least @1@

[Description]

Lack of max_prepared_transactions.

[System Processing]

Processing will be aborted.

[Action]

Please increase max_prepared_transactions.

message contents do not agree with length in message type "@1@"

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

missing "=" after "@1@" in connection info string

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

new primary COORDINATOR has been connected

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

no cmdata directory specified and environment variable CMDATA unset

[Description]

No cmdata directory specified and environment variable CMDATA unset.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

no connection manager directory is specified and environment variable CMDATA is unset

[Description]

No connection manager directory is specified and environment variable CMDATA is unset.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

no data left in message

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

no empty local buffer available

[Description]

There was insufficient free space in the disk of the database server during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Delete user data stored in the database server to free up space on the disk.

no operation mode is specified

[Description]

No operation mode is specified.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

no target server address from Connection Manager

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

node_name is needed for instance #@1@

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

---

node_name is not needed for instance #@1@

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

---

not enough elements in RWConflictPool to record a potential read/write conflict

[Description]

There was insufficient free space in the server's memory during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Estimate memory usage and take the following action:

- If the number of simultaneous connections from client applications is high, reduce it.

- If the number of simultaneous SQL executions is high, reduce it.

---

not enough elements in RWConflictPool to record a read/write conflict

[Description]

There was insufficient free space in the server's memory during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Estimate memory usage and take the following action:

- If the number of simultaneous connections from client applications is high, reduce it.

- If the number of simultaneous SQL executions is high, reduce it.

---

not enough shared memory for data structure "@1@" (@2@ bytes requested)

[Description]

There was insufficient free space in the database server's shared memory during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Estimate memory usage and take the following action:

- If the number of simultaneous connections from client applications is high, reduce it.

- If the number of simultaneous SQL executions is high, reduce it.

---

number of workers must not be negative

[Description]

The number of workers must not be negative.

[System Processing]

Processing will be aborted.

[Action]

Please specify more than one.

---

oldest multixact is far in the past

[Description]

Terminated normally but a warning was output.

[System Processing]

Continues processing.

[Action]

Close open transactions with multixacts soon to avoid wraparound problems.

---

one of -d/--dbname and -f/--file must be specified

[Description]

one of -d/--dbname and -f/--file must be specified

[System Processing]

Processing will be aborted.

[Action]

Check the command-line, and re-execute the command with correct options.

---

out of memory

[Description]

There was insufficient free space in the server's memory during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Estimate memory usage and take the following action:

- If the number of simultaneous connections from client applications is high, reduce it.

- If the number of simultaneous SQL executions is high, reduce it.

---

out of memory allocating GSSAPI buffer (@1@)

[Description]

There was insufficient free space in the client's memory during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Estimate memory usage and take the following action:

- Modify the application to reduce memory usage.

---

out of memory for query result

[Description]

There was insufficient free space in the client's memory during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Estimate memory usage and take the following action:

- Modify the application to reduce memory usage.

---

out of memory on line @1@

[Description]

There was insufficient free space in the server's memory during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Estimate memory usage and take the following action:

- If the number of simultaneous connections from client applications is high, reduce it.

- If the number of simultaneous SQL executions is high, reduce it.

---

out of shared memory

[Description]

There was insufficient free space in the database server's shared memory during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Estimate memory usage and take the following action:

- If the number of simultaneous connections from client applications is high, reduce it.

- If the number of simultaneous SQL executions is high, reduce it.

---

pgx_global_metacache must be set to at least 10MB when enabled.

[Description]

pgx_global_metacache must be set to at least 10MB when enabled.

[System Processing]

Processing will be aborted.

[Action]

Set the pgx_global_metacache parameter to at least 10MB and restart the database.

pgx_loader only available using COPY FROM

[Description]

pgx_loader can be available only COPY FROM command.

[System Processing]

Processing will be aborted.

[Action]

Please specify COPY FROM statement.

pid file "@1@/conmgr.pid" does not exist

[Description]

Pid file does not exist.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

posting list is too long

[Description]

An error occurred during execution of the application or command.

[System Processing]

Processing will be aborted.

[Action]

Reduce the value of maintenance_work_mem.

postmaster exited during a parallel loading

[Description]

postmaster exited during a parallel loading.

[System Processing]

Processing will be aborted.

[Action]

Please refer to the server log, and determine the cause of the error.

pre-existing shared memory block (key @1@, ID @2@) is still in use

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

private key file "@1@" has group or world access; file must have permissions u=rw (0600) or less if owned by the current user, or permissions u=rw,g=r (0640) or less if owned by root

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

private key file "@1@" is not a regular file

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

private key file "@1@" must be owned by the database user or root

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

protocol error: id=0x@1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

query failed: @1@

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

query would be affected by row-level security policy for table "@1@"

[Description]

An error occurred during execution of the application or command.

[System Processing]

Processing will be aborted.

[Action]

To disable the policy for the table owner, use ALTER TABLE NO FORCE ROW LEVEL SECURITY.

real and effective user IDs must match

[Description]

Real and effective user IDs must match.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

realm name too long

[Description]

An error occurred while translating domain name to Kerberos realm name in SSL.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the domain name is written correctly.

received fast shutdown request

[Description]

Received fast shutdown request.

[System Processing]

Processing continues.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

received immediate shutdown request

[Description]

Received immediate shutdown request.

[System Processing]

Processing continues.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

received invalid response to SSL negotiation: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

received invalid startup packet

[Description]

Received invalid startup packet.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

received smart shutdown request

[Description]

Received smart shutdown request.

[System Processing]

Processing continues.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

received unencrypted data after GSSAPI encryption request

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

received unencrypted data after SSL request

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

recovery has paused

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

rejected a connection from application due to going shutdown

[Description]

Rejected a connection from application due to going shutdown.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

remote node closes heartbeat connection

[Description]

Remote node closes heartbeat connection.

[System Processing]

Processing aborts.

[Action]

Check log of remote system.

remote node shuts down heartbeat connection

[Description]

Remote node shuts down heartbeat connection.

[System Processing]

Processing aborts.

[Action]

Check log of remote system.

Remove those directories, or set allow_in_place_tablespaces to ON transiently to let recovery complete.

[Description]

An error occurred due to unexpected directory entry found at specific path

[System Processing]

Cancels processing.

[Action]

Remove those unexpected directories, or set allow_in_place_tablespaces to ON transiently to let recovery complete.

results of copying file does not exist

[Description]

The backup data may be corrupted.

[System Processing]

Processing will be aborted.

[Action]

Copy the backup data to the backup storage directory from backup media.

RETURNING list entry has type @1@, but column has type @2@.

[Description]

Supplementary information was output.

[System Processing]

None.

[Action]

Refer to this message together with the message that was output immediately beforehand.

role with OID @1@ does not exist

[Description]

An error occurred during DB Server processing in the database.

[System Processing]

Processing will be aborted.

[Action]

Refer to this message together with the message that was output immediately beforehand.

root certificate file "@1@" does not existEither provide the file or change sslmode to disable server certificate verification.

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

SELECT target entry has type @1@, but column has type @2@.

[Description]

Supplementary information was output.

[System Processing]

None.

[Action]

Refer to this message together with the message that was output immediately beforehand.

SELECT target entry is named "@1@".

[Description]

Supplementary information was output.

[System Processing]

None.

[Action]

Refer to this message together with the message that was output immediately beforehand.

select() failed in postmaster: @1@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

server certificate for "@1@" does not match host name "@2@"

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

server sent data ("D" message) without prior row description ("T" message)

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

service file "@1@" not found

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

set-valued function called in context that cannot accept a set

[Description]

Set-valued function called in context that cannot accept a set.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

Sleep until another process releases global meta cache because global meta cache cannot be swept away yet. Please increase pgx_global_metacache.

[Description]

Sleep until another process releases global meta cache because global meta cache cannot be swept away yet. Please increase pgx_global_metacache.

[System Processing]

Processing continues.

[Action]

If this message occurs frequently, increase the pgx_global_metacache parameter setting and restart the database.

SSL certificate's common name contains embedded null

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

SSL connection has been closed unexpectedly

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

SSL error: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

stack depth limit exceeded

[Description]

The depth of the execution stack exceeded the allowable value during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

If executing a complex SQL, modify it to a simple SQL.

standby mode is not supported by single-user servers

[Description]

An error occurred during DB Server processing in the database.

[System Processing]

Processing will be aborted.

[Action]

Refer to this message together with the message that was output immediately beforehand.

started conmgr process successfully

[Description]

Started conmgr process successfully.

[System Processing]

Continues processing.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

starting conmgr process

[Description]

Starting conmgr process.

[System Processing]

Continues processing.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

stopped conmgr process successfully

[Description]

Stopped conmgr process successfully.

[System Processing]

Continues processing.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

String constants with Unicode escapes cannot be used when standard_conforming_strings is off.

[Description]

Supplementary information was output.

[System Processing]

None.

[Action]

Refer to this message together with the message that was output immediately beforehand.

string is not a valid identifier: "@1@"

[Description]

An error occurred during execution of the application or command.

[System Processing]

Processing will be aborted.

[Action]

The string may have unclosed double quotation marks.

synchronous_standby_names parser failed

[Description]

An error occurred during execution of the application or command.

[System Processing]

Processing will be aborted.

[Action]

Check the value of synchronous_standby_names, which may come from postgresql.conf or the SQL command line.

syntax error in service file "@1@", line @2@

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

syntax error near '@1@' in configuration file "@2@" line @3@

[Description]

Syntax error in configuration file.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

syntax error near end of line in configuration file "@1@" line @2@

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

TDE_h extension must be loaded via shared_preload_libraries

[Description]

The TDE_h extension cannot be dynamically loaded by the user.

[System Processing]

Processing will be aborted.

[Action]

In postgresql.conf specify shared_preload_libraries = 'tde_h'

TDE_h: @1@ failed: @2@ [@3@]

[Description]

The PKCS11 function call failed.

[System Processing]

Processing will be aborted.

[Action]

The specified value in the PKCS11 configuration file (grep11client.yaml) or the TDE_h execution environment is incorrect, or the HPCS is detecting an error. Check the message and take appropriate action according to the reported PKCS11 error code. Also, if you have specified a log in the PKCS11 configuration file, check the output log and take action.

TDE_h: C_Login: @1@ [@2@]

[Description]

The call to the C_OpenSession function of PKCS11 has failed.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and take appropriate action based on the reported PKCS11 error code.

TDE_h: Expected key not found: @1@ [@2@]

[Description]

During encryption or decryption processing TDE_h was not able to find an expected PKCS11 key.

[System Processing]

Processing will be aborted.

[Action]

The keystore and the PKCS11 token key objects are not consistent with each other. This may happen if the keystore and/or PKCS11 token directory have been incorrectly copied (or not copied at all). Check all files have been copied correctly.

TDE_h: Operation is not supported

[Description]

The TDE_h extension cannot be dynamically loaded by the user.

[System Processing]

Processing will be aborted.

[Action]

In postgresql.conf specify shared_preload_libraries = 'tde_h'

TDE_h: HPCS Service API Key is not available at load time

[Description]

The PKCS11 API key associated with the token of Slot ID was not defined at load time.

[System Processing]

This is only a warning message when TDE_h is loaded.

[Action]

The token API key must be known to tde_h before secure keys can be used. Specify the API key at load time with --api-key option, or by GUC tde_h.API_KEY in postgresql.conf. Or, specify the API key at run-time using SELECT pgx_open_keystore(API Key); or SELECT pgx_set_master_key(API Key);

TDE_h: PKCS11 Slot ID is not defined

[Description]

The PKCS11 Slot ID for transparent data encryption is not defined.

[System Processing]

Processing will be aborted.

[Action]

In postgresql.conf set the GUC tded_h.SLOT_ID to the PKCS11 Slot ID configured for transparent data encryption.

TDE_h: Slot @1@ C_GetMechanismInfo CKM_AES_CBC: @2@ [@3@]

[Description]

The specified Slot ID does not support a mechanism required by TDE_h.

[System Processing]

Processing will be aborted.

[Action]

Specify a different token which supports the necessary mechanism.

TDE_h: Slot @1@ C_GetMechanismInfo CKM_AES_KEY_GEN: @2@ [@3@]

[Description]

The specified Slot ID does not support a mechanism required by TDE_h.

[System Processing]

Processing will be aborted.

[Action]

Specify a different token which supports the necessary mechanism.

TDE_h: Slot @1@ C_OpenSession: @2@ [@3@]

[Description]

The call to the C_OpenSession function of PKCS11 has failed.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and take appropriate action based on the reported PKCS11 error code.

TDE_h: Slot @1@ mechanism does not support CKF_DECRYPT

[Description]

The specified Slot ID does not support a mechanism required by TDE_h.

[System Processing]

Processing will be aborted.

[Action]

Specify a different token which supports the necessary mechanism.

TDE_h: Slot @1@ mechanism does not support CKF_ENCRYPT

[Description]

The specified Slot ID does not support a mechanism required by TDE_h.

[System Processing]

Processing will be aborted.

[Action]

Specify a different token which supports the necessary mechanism.

TDE_h: Slot @1@ mechanism does not support CKF_GENERATE

[Description]

The specified Slot ID does not support a mechanism required by TDE_h.

[System Processing]

Processing will be aborted.

[Action]

Specify a different token which supports the necessary mechanism.

TDE_h: Slot @1@ token model '@2@' is not supported

[Description]

The PKCS11 token model at the specified Slot ID is not suported by TDE_h.

[System Processing]

Processing will be aborted.

[Action]

Only "EP11" and "CCA" tokens are supported.

TDE_h: Slot @1@ token model is '@2@'

[Description]

Logs the PKCS11 Slot ID and token model that TDE_h will be using.

[System Processing]

This is an information messaage when TDE_h is loaded.

[Action]

No action is required. Confirm the Slot and token type are as expected.

TDE_h: Unable to initialize for TDE_h environment

[Description]

Failed to initialize execution environment for TDE_h.

[System Processing]

Processing will be aborted.

[Action]

The specified value in the PKCS11 configuration file (grep11client.yaml) or the TDE_h execution environment is incorrect, or the HPCS is detecting an error. Check the message and take appropriate action according to the reported PKCS11 error code. Also, if you have specified a log in the PKCS11 configuration file, check the output log and take action.

TDE_h: Unable to initialize the PKCS11 library: 0x@1@ [@2@]

[Description]

The call to the C_Initalize function of PKCS11 has failed.

[System Processing]

Processing will be aborted.

[Action]

The specified value in the PKCS11 configuration file (grep11client.yaml) or the TDE_h execution environment is incorrect, or the HPCS is detecting an error. Check the message and take appropriate action according to the reported PKCS11 error code. Also, if you have specified a log in the PKCS11 configuration file, check the output log and take action.

TDEz: Operation is not supported

[Description]

The TDE_z extension cannot be dynamically loaded by the user.

[System Processing]

Processing will be aborted.

[Action]

In postgresql.conf specify shared_preload_libraries = 'TDE_z'

TDE_z extension must be loaded via shared_preload_libraries

[Description]

The TDE_z extension cannot be dynamically loaded by the user.

[System Processing]

Processing will be aborted.

[Action]

In postgresql.conf specify shared_preload_libraries = 'TDE_z'

TDE_z: C_Login: @1@ [@2@]

[Description]

The call to the C_OpenSession function of PKCS#11 has failed.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and take appropriate action based on the reported PKCS#11 error code.

TDE_z: Expected key not found: @1@ [@2@]

[Description]

During encryption or decryption processing TDE_z was not able to find an expected opencryptoki key.

[System Processing]

Processing will be aborted.

[Action]

The keystore and the opencryptoki token key objects are not consistent with each other. This may happen if the keystore and/or opencryptoki token directory have been incorrectly copied (or not copied at all). Check all files have been copied correctly.

TDE_z: PKCS11 Slot ID is not defined

[Description]

The opencryptoki Slot ID for transparent data encryption is not defined.

[System Processing]

Processing will be aborted.

[Action]

In postgresql.conf set the GUC TDE_z.SLOT_ID to the opencryptoki Slot ID configured for transparent data encryption.

TDE_z: PKCS11 User PIN is not available at load time

[Description]

The opencryptoki User PIN associated with the token of Slot ID was not defined at load time.

[System Processing]

This is only a warning message when TDE_z is loaded.

[Action]

The token User PIN must be known to TDE_z before secure keys can be used. Specify the User PIN at load time with --user-pin option, or by GUC TDE_z.USER_PIN in postgresql.conf. Or, specify the User PIN at run-time using SELECT pgx_open_keystore(user pin); or SELECT pgx_set_master_key(user pin);

TDE_z: Slot @1@ C_GetMechanismInfo CKM_AES_CBC: @2@ [@3@]

[Description]

The specified Slot ID does not support a mechanism required by TDE_z.

[System Processing]

Processing will be aborted.

[Action]

Specify a different token which supports the necessary mechanism.

TDE_z: Slot @1@ C_GetMechanismInfo CKM_AES_KEY_GEN: @2@ [@3@]

[Description]

The specified Slot ID does not support a mechanism required by TDE_z.

[System Processing]

Processing will be aborted.

[Action]

Specify a different token which supports the necessary mechanism.

TDE_z: Slot @1@ C_OpenSession: @2@ [@3@]

[Description]

The call to the C_OpenSession function of PKCS#11 has failed.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and take appropriate action based on the reported PKCS#11 error code.

TDE_z: Slot @1@ mechanism does not support CKF_DECRYPT

[Description]

The specified Slot ID does not support a mechanism required by TDE_z.

[System Processing]

Processing will be aborted.

[Action]

Specify a different token which supports the necessary mechanism.

TDE_z: Slot @1@ mechanism does not support CKF_ENCRYPT

[Description]

The specified Slot ID does not support a mechanism required by TDE_z.

[System Processing]

Processing will be aborted.

[Action]

Specify a different token which supports the necessary mechanism.

TDE_z: Slot @1@ mechanism does not support CKF_GENERATE

[Description]

The specified Slot ID does not support a mechanism required by TDE_z.

[System Processing]

Processing will be aborted.

[Action]

Specify a different token which supports the necessary mechanism.

TDE_z: Slot @1@ token model '@2@' is not supported

[Description]

The opencryptoki token model at the specified Slot ID is not supported by TDE_z.

[System Processing]

Processing will be aborted.

[Action]

Only "EP11" and "CCA" tokens are supported.

TDE_z: Slot @1@ token model is '@2@'

[Description]

Logs the opencryptoki Slot ID and token model that TDE_z will be using.

[System Processing]

This is an information messaage when TDE_z is loaded.

[Action]

No action is required. Confirm the Slot and token type are as expected.

TDE_z: Unable to initialize the opencryptoki library: 0x@1@ [@2@]

[Description]

The call to the C_Initalize function of PKCS#11 has failed.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and take appropriate action based on the reported PKCS#11 error code.

TDE-type is inconsistent between keystore and postgresql.conf.

[Description]

The TDE type specified in shared_preload_libraries in postgresql.conf does not match the keystore format of the execution environment.

[System Processing]

Processing will be aborted.

[Action]

Inconsistent value specified for shared_preload_libraries in keystore and postgresql.conf. Determine which is correct and take appropriate action.

---

terminating connection due to conflict with recovery

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

---

terminating connection due to idle-session timeout

[Description]

Timeout occurred during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Check the following:

- If executing SQL that outputs a large volume of search results, add a conditional expression to filter the results further.

- If numerous SQLs are being simultaneously executed, reduce the number of simultaneously executed SQLs.

- If a large volume of data is to be updated in a single transaction, modify the SQL to reduce the volume of data to be updated in a single transaction.

- If executing a complex SQL, modify it to a simple SQL.

- Check if there are any problems in the network.

- Before conducting maintenance that involves the processing of a large volume of data, use the SET statement to temporarily increase the value of maintenance_work_mem.

---

terminating connection due to unexpected postmaster exit

[Description]

An error occurred during execution of the application or command.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the server is still running.

terminating walsender process due to replication timeout

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

terminator: could not prepare statement for "@1@"

[Description]

Could not prepare statement.

[System Processing]

Processing aborts.

[Action]

Check messages near this message.

terminator: terminated connections: ipaddress="@1@"

[Description]

Terminated connections.

[System Processing]

Processing continues.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

the database system is in recovery mode

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

the database system is not accepting connections

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

the database system is not yet accepting connections

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

the database system is shutting down

[Description]

The database system is shutting down.

[System Processing]

Processing will be aborted.

[Action]

This message is output when the stopping process is operating normally. Retry any necessary applications or commands after restarting the database system.

the database system is starting up

[Description]

The database system is starting up.

[System Processing]

Processing will be aborted.

[Action]

This message is output when the startup process or recovery process is operating normally. Restart any necessary the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

The program "@1@" is needed by @2@ but was not found in thesame directory as "@3@".Check your installation.

[Description]

A required program was not found. Check your installation.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

The program "@1@" was found by "@2@"but was not the same version as @3@.Check your installation.

[Description]

The program was not the same version.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

The sslservercertcn "@1@" could not be verified

[Description]

value of sslservercertcn is different from common name in the server certificate.

[System Processing]

Processing is aborted.

[Action]

Set SSL certificate's common name to sslservercertcn.

timeout expired

[Description]

Timeout occurred during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Check the following:

- If executing SQL that outputs a large volume of search results, add a conditional expression to filter the results further.

- If numerous SQLs are being simultaneously executed, reduce the number of simultaneously executed SQLs.

- If a large volume of data is to be updated in a single transaction, modify the SQL to reduce the volume of data to be updated in a single transaction.

- If executing a complex SQL, modify it to a simple SQL.

- Check if there are any problems in the network.

- Before conducting maintenance that involves the processing of a large volume of data, use the SET statement to temporarily increase the value of maintenance_work_mem.

too long literal in configuration file "@1@" line @2@

[Description]

Too long literal in configuration file.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

too many background workers

[Description]

Up to max_worker_processes background worker can be registered with the current settings.

[System Processing]

Processing will be aborted.

[Action]

Consider increasing the value of configuration parameter max_worker_processes.

---

too many command-line arguments (first is "@1@")

[Description]

Too many command-line arguments.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

---

too many grouping sets present (maximum 4096)

[Description]

An error occurred during execution of the application or command.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the items in the GROUPING SET are less than the maximum allowable value.

---

translated account name too long

[Description]

An error occurred while translating account name to Kerberos user name in SSL.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the account name is written correctly.

---

trigger created with DO block cannot be replaced by EXECUTE PROCEDURE

[Description]

Trigger created with DO block cannot be replaced by EXECUTE PROCEDURE.

[System Processing]

Processing is aborted.

[Action]

Please redefine it after deleting the trigger.

---

trigger created with EXECUTE PROCEDURE cannot be replaced by DO block

[Description]

Trigger created with EXECUTE PROCEDURE cannot be replaced by DO block.

[System Processing]

Processing is aborted.

[Action]

Please redefine it after deleting the trigger.

---

type input function @1@ should not be volatile

[Description]

An error occurred during execution of the application or command.

[System Processing]

Continues processing.

[Action]

Check the message text and confirm that functions are not marked as volatile.

---

type modifier input function @1@ should not be volatile

[Description]

An error occurred during execution of the application or command.

[System Processing]

Continues processing.

[Action]

Check the message text and confirm that functions are not marked as volatile.

---

type modifier output function @1@ should not be volatile

[Description]

An error occurred during execution of the application or command.

[System Processing]

Continues processing.

[Action]

Check the message text and confirm that functions are not marked as volatile.

---

type output function @1@ should not be volatile

[Description]

An error occurred during execution of the application or command.

[System Processing]

Continues processing.

[Action]

Check the message text and confirm that functions are not marked as volatile.

---

type receive function @1@ should not be volatile

[Description]

An error occurred during execution of the application or command.

[System Processing]

Continues processing.

[Action]

Check the message text and confirm that functions are not marked as volatile.

type send function @1@ should not be volatile

[Description]

An error occurred during execution of the application or command.

[System Processing]

Continues processing.

[Action]

Check the message text and confirm that functions are not marked as volatile.

unexpected directory entry "@1@" found in @2@

[Description]

Unexpected directory entry found at specific path while recovery.

[System Processing]

Cancels processing.

[Action]

Remove those unexpected directories, or set allow_in_place_tablespaces to ON transiently to let recovery complete.

unexpected EOF on standby connection

[Description]

An error occurred because execution is temporarily impossible.

[System Processing]

Processing will be aborted.

[Action]

Restart the application. If the same error occurs when you restart the application, to check if there are any problems in the database server.

unexpected EOF within message length word

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

unexpected field count in "D" message

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

unexpected message from server during startup

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

unexpected message type "@1@"

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

unexpected message type 0x@1@ during COPY from stdin

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

unexpected response from server; first received character was "@1@"

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

Unicode escapes must be \\uXXXX or \\UXXXXXXXX.

[Description]

Supplementary information was output.

[System Processing]

None.

[Action]

Refer to this message together with the message that was output immediately beforehand.

unrecognized operation mode "@1@"

[Description]

Unrecognized operation mode was specified.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

unrecognized SSL error code: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

unrecognized stop mode "@1@"

[Description]

Unrecognized stop mode was specified.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

unrecognized value "@1@" for option '-i'

[Description]

Unrecognized value specified for option '-i'.

[System Processing]

Processing aborts.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

unterminated quoted string in connection info string

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

Use '' to write quotes in strings, or use the escape string syntax (E'...').

[Description]

Supplementary information was output.

[System Processing]

None.

[Action]

Refer to this message together with the message that was output immediately beforehand.

Use '' to write quotes in strings. \\' is insecure in client-only encodings.

[Description]

Supplementary information was output.

[System Processing]

None.

[Action]

Refer to this message together with the message that was output immediately beforehand.

Use the escape string syntax for backslashes, e.g., E'\\\\'.

[Description]

Supplementary information was output.

[System Processing]

None.

[Action]

Refer to this message together with the message that was output immediately beforehand.

Use the escape string syntax for escapes, e.g., E'\\r\'.

[Description]

Supplementary information was output.

[System Processing]

None.

[Action]

Refer to this message together with the message that was output immediately beforehand.

waiting conmgr process to connect to watchdog

[Description]

Waiting conmgr process to connect to watchdog.

[System Processing]

Continues processing.

[Action]

Recheck command line, configuration file, or logic of your application. Hint message may follow with this message.

watchdog: could not accept heartbeat connection

[Description]

Could not accept heartbeat connection.

[System Processing]

Processing aborts.

[Action]

Check messages near this message.

watchdog: could not send attribute

[Description]

Could not send attribute.

[System Processing]

Processing aborts.

[Action]

Check messages near this message.

You might need to increase max_worker_processes.

[Description]

You might need to increase max_worker_processes.

[System Processing]

Processing will be aborted.

[Action]

You might need to increase max_worker_processes

An error occurred while setting up the GSS Encoded connection.

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

An error occurred while setting up the SSL connection.

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

An I/O error occurred while sending to the backend.

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

An unexpected result was returned by a query.

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

Cannot connect to Connection Manager when targetServerType is "preferPrimary".

[Description]

An error occurred during execution of the application or command.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the application is written correctly and the command is being used correctly.

CommandComplete expected COPY but got:

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

Connection attempt timed out.

[Description]

Timeout occurred during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Check the following:

- If executing SQL that outputs a large volume of search results, add a conditional expression to filter the results further.

- If numerous SQLs are being simultaneously executed, reduce the number of simultaneously executed SQLs.

- If a large volume of data is to be updated in a single transaction, modify the SQL to reduce the volume of data to be updated in a single transaction.

- If executing a complex SQL, modify it to a simple SQL.

- Check if there are any problems in the network.

- Before conducting maintenance that involves the processing of a large volume of data, use the SET statement to temporarily increase the value of maintenance_work_mem.

Could not close SSL certificate file @1@.

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

Could not find a server with specified targetServerType: @1@

[Description]

Could not find a suitable target server.

[System Processing]

Processing is aborted.

[Action]

Check following settings (host, IP address, port number,or targetServer):

- Connection string

- Connection service file

- Data source of JDBC or ODBC

- Environment variables for default connection parameter values(ex. PGHOST)

- Arguments of functions of libpq

- Options of command

Database cannot be null

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

Database connection failed when canceling copy operation

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

Database connection failed when ending copy

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

Database connection failed when reading from copy

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

Database connection failed when starting copy

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

Database connection failed when writing to copy

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

---

enableFdwAcs cannot be set "on" when targetServerType is "preferPrimary".

[Description]

An error occurred during execution of the application or command.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the application is written correctly and the command is being used correctly.

Expected an EOF from server, got: @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

Expected command status BEGIN, got @1@.

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

Got @1@ error responses to single copy cancel request

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

GSS Authentication failed

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

hstore key must not be null

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

Interrupted while attempting to connect.

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

Invalid gssEncMode value: @1@

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

Invalid targetServerType value: @1@

[Description]

value of targetserver is invalid.

[System Processing]

Processing is aborted.

[Action]

Set one of the following:

- primary

- standby

- prefer_standby

- any(can be specified only JDBC)

Missing expected error response to copy cancel request

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

Neither Subject.doAs (Java before 18) nor Subject.callAs (Java 18+) method found

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

One or more ClientInfo failed.

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

PreparedStatement can have at most @1@ parameters. Please consider using arrays, or splitting the query in several ones, or using COPY. Given query has @2@ parameters

[Description]

An error occurred during execution of the application or command.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the application is written correctly and the command is being used correctly.

Protocol error. Session setup failed.

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

Ran out of memory retrieving query results.

[Description]

There was insufficient free space in the server's memory during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Estimate memory usage and take the following action:

- If the number of simultaneous connections from client applications is high, reduce it.

- If the number of simultaneous SQL executions is high, reduce it.

Read from copy failed.

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

The connection attempt failed.

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

The server does not support GSS Encoding.

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

The server does not support GSS Encryption.

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

The server requested password-based authentication, but no password was provided by plugin @1@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

The server requested SCRAM-based authentication, but the password is an empty string.

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

The server''s client_encoding parameter was changed to @1@. The JDBC driver requires client_encoding to be UTF8 for correct operation.

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

The server''s DateStyle parameter was changed to @1@. The JDBC driver requires DateStyle to begin with ISO for correct operation.

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

The server''s standard_conforming_strings parameter was reported as @1@. The JDBC driver expected on or off.

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

The sslservercertcn @1@ could not be verified by hostnameverifier @2@.

[Description]

value of sslservercertcn is different from common name in the server certificate.

[System Processing]

Processing is aborted.

[Action]

- Set SSL certificate's common name to sslservercertcn.

- Check the program of class specified by hostnameverifier.

Unable to find pkcs12 keystore.

[Description]

The operating environment such as the status of the connection definition file and the connection method specification is abnormal.

[System Processing]

Processing will be aborted.

[Action]

Confirm that the operating environment such as the status of the connection definition file and the connection method specification is normal.

Unable to load Authentication Plugin @1@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

Unable to parse URL @1@

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

Unexpected command status: @1@.

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

Unexpected copydata from server for @1@

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

Unknown Response Type @1@.

[Description]

An error occurred during communication between the application and the database server.

[System Processing]

Processing will be aborted.

[Action]

Check if there are any problems in the network, eliminate the cause of any error and re-execute the command.

User cannot be null

[Description]

The database server was disconnected during execution of the application.

[System Processing]

Processing will be aborted.

[Action]

Communication may have been disconnected for the following reasons:

- An error occurred in the communication line (TCP/IP etc.)

- The database server terminated abnormally.

Take the following actions:

- Eliminate the cause of the communication disconnection.

Examine the application and check whether the transaction for implementing update is a long transaction. Judge whether it is a long transaction from the following viewpoints and modify the application.

a) If the COMMIT process is not executed after update, add the COMMIT process.

b) If the total number of update records in a single transaction is high, split it into short transactions.

c) If search was conducted for a long period of time after update, execute COMMIT after update or review the search SQL statement.

- If none of the above situations applies, perform the following:

a) Confirm that the database server has not stopped.

b) If the database server is starting or stopping, re-execute the command after the database server starts.

Value is not an OID: @1@

[Description]

An error occurred during execution of the application or command.

[System Processing]

Processing will be aborted.

[Action]

Check the message text and confirm that the application is written correctly and the command is being used correctly.

Chapter 3 Mirroring Controller Messages

This chapter explains messages output by Mirroring Controller.

3.1 Message Numbers Beginning with MCA00000

3.1.1 MCA00001

could not read file "{0}": exception={1}: {2}

[Description]

Could not read the file.

[Parameters]

{0}: file name

{1}: exception type

{2}: exception detail

[System Processing]

Processing will be aborted.

[Action]

Identify the cause according to the message, and then remove it.

3.1.2 MCA00002

{0}: wrong number of server ID in definition file "{1}"

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: command name

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.1.3 MCA00003

{0}: server ID not found in definition file "{1}" line {2}

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: line number

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.1.4 MCA00004

{0}: server ID specified in definition file "{1}" too long (max {2} bytes) line {3}

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: max length of server ID

{3}: line number

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.1.5 MCA00005

invalid host name or IP address "{1}" in definition file "{0}"

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: file name

{1}: host name or IP address

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.1.6 MCA00006

{0}: invalid port number in definition file "{1}" line {2}

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: line number

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.1.7 MCA00007

{0}: invalid format specified in definition file "{1}" line {2}

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: line number

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.1.8 MCA00008

{0}: invalid value for parameter "{2}" in definition file "{1}"

[Description]

Invalid parameter was found in definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: parameter name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.1.9 MCA00009

{0}: no value for parameter "{2}" specified in definition file "{1}"

[Description]

Invalid parameter was found in definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: parameter name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.1.10 MCA00010

{0}: unrecognized parameter "{2}" in definition file "{1}"

[Description]

Unrecognized parameter was found in definition file.

[Parameters]

{0}: file name

{1}: command name

{2}: parameter name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.1.11 MCA00011

%s: no operation mode specified

[Description]

No operation mode was specified.

[Parameters]

%s: command name

[System Processing]

Processing will be aborted.

[Action]

Re-execute after specifying operation modes.

3.1.12 MCA00012

%s: unrecognized operation mode "%s"

[Description]

Unrecognized operation mode was specified.

[Parameters]

%s: command name

%s: operation mode

[System Processing]

Processing will be aborted.

[Action]

Re-execute after specifying operation modes.

3.1.13 MCA00013

%s: option "%s" duplicated

[Description]

Certain option is duplicated.

[Parameters]

%s: command

%s: option

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting options.

3.1.14 MCA00014

%s: "%s" option conflicts with "%s" option

[Description]

Options are conflicting.

[Parameters]

%s: command

%s: option

%s: option

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting options.

3.1.15 MCA00015

%s: option requires an argument -- %s

[Description]

No argument specified for the option.

[Parameters]

%s: command

%s: option

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting options.

3.1.16 MCA00016

%s: neither "%s" option nor environment variable "%s" specified

[Description]

Both of required option and equivalent environment variable were not specified.

[Parameters]

%s: command

%s: option

%s: environment variable

[System Processing]

Processing will be aborted.

[Action]

Re-execute after specifying required option or equivalent environment variable.

3.1.17 MCA00017

%s: out of memory

[Description]

Out of memory error occurred.

[Parameters]

%s: command

[System Processing]

Processing will be aborted.

[Action]

Obtain free memory space by stopping unnecessary processes or changing system settings.

3.1.18 MCA00018

out of memory

[Description]

Out of memory error occurred.

[System Processing]

Processing will be aborted.

[Action]

Obtain free memory space by stopping unnecessary processes or changing system settings.

3.1.19 MCA00019

detected an error on the monitored object "{0}({1})": {2}

[Description]

An error was detected on the monitored object.

[Parameters]

{0}: monitored object (server, database process, data storage destination directory, transaction log storage destination directory, tablespace directory)

{1}: object name (host name, database process name, directory path)

{2}: error detail ("no response:", "read/write error:" and detailed information)

[System Processing]

Perform failover or detaching.

If failover or detaching is already performed, these functions would be disabled.

[Action]

Refer to the description about workaround for failure of "Cluster Operation Guide (Database Multiplexing)".

3.1.20 MCA00020

unexpected error occurred in the monitoring process: {0}

[Description]

Monitoring process cannot continue because an unexpected error was occurred during its processing.

[Parameters]

{0}: error detail

[System Processing]

Continues processing.

[Action]

Refer to the description about workaround for failure of "Cluster Operation Guide (Database Multiplexing)".

3.1.21 MCA00021

starting to {2} from {0} to {1}.

[Description]

Switching standby server to primary server.

[Parameters]

{0}: server ID of primary server

{1}: server ID of standby server

{2}: "fail over"or"switch over"

[Action]

If Mirroring Controller executed switching automatically, find the message output before this message from system log or event log to identify the cause of switching, and then work around according to the Action of the message.

3.1.22 MCA00022

{2} completed.switched from {0} to {1}

[Description]

Switching standby server to primary server was completed.

[Parameters]

{0}: server ID of primary server

{1}: server ID of standby server

{2}: "fail over"or"switch over"

3.1.23 MCA00023

failed to {2} from {0} to {1}

[Description]

Failed to switch to primary server because of unexpected failure.

[Parameters]

{0}: server ID of primary server

{1}: server ID of standby server

{2}: "fail over"or"switch over"

[System Processing]

Processing of switching will be aborted.

[Action]

Find the message output before this message from display, system log or event log, and then work around according to the Action of the message.

3.1.24 MCA00024

starting to detach standby server "{0}" {1}

[Description]

Detaching standby server because of failure detected on standby server.

[Parameters]

{0}: server ID

{1}: "automatically" or none

[Action]

If Mirroring Controller executed detaching automatically, find the message output before this message from system log or event log to identify the cause of detaching, and then work around according to the Action of the message.

3.1.25 MCA00025

detach standby server "{0}" completed {1}

[Description]

Detached standby server.

[Parameters]

{0}: server ID

{1}: "automatically" or none

3.1.26 MCA00026

failed to {2} standby server "{0}" {1}

[Description]

Failed to detach standby server.

[Parameters]

{0}: server ID

{1}: "automatically" or none

{2}: "detach"

[System Processing]

Processing of detaching will be aborted.

[Action]

Refer to the description about workaround for failure of "Cluster Operation Guide (Database Multiplexing)".

3.1.27 MCA00027

another "{0}" command is running

[Description]

Cannot execute command with this operation mode because another command is running on the same or another server.

[Parameters]

{0}: command name

[System Processing]

Processing will be aborted.

[Action]

There is a case executing another command. Wait for completion of another command on the same or another server, and then re-execute.

In addition, there are the following cases when using mc_ctl command.

There is a case under processing of a failover and an automatic detach by Mirroring Controller. Wait for completion of the processing under operation, and re-execute.

If any of the following cases occurs, there is a possibility that the processing of Mirroring Controller interrupts. Re-execute the mc_ctl command after restarting Mirroring Controller.

- When abnormality occurs in the network

- When another server is downed

- When Mirroring Controller is stopped forcibly

3.1.28 MCA00028

communication timeout of Mirroring Controller occurred server:"{0}"

[Description]

Either of the followings has occurred.

- communication timeout between mc_ctl command and Mirroring Controller process has occurred.

- communication timeout between Mirroring Controller processes have occurred.

- Terminating database instance was not completed in the specified time.

[Parameters]

{0}: server ("localhost"or server ID)

[System Processing]

Processing will be aborted.

[Action]

Completion synchronization for terminating database instance might have timed out, because connections remained. Disconnect all connections, and re-execute it.

Reduce CPU or network load caused by the other processes. If could not reduce it, extend remote_call_timeout in "server identifier".conf.

3.1.29 MCA00029

could not create PID file "{0}" of Mirroring Controller detail of cause:"{1}"

[Description]

Could not create PID file of Mirroring Controller.

[Parameters]

{0}: file name

{1}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Identify the cause according to the message, and then remove it.

3.1.30 MCA00030

could not remove PID file "%s" of Mirroring Controller detail of cause:"%s"

[Description]

Could not remove PID file of Mirroring Controller.

[Parameters]

%s: file name

%s: detail of cause

[Action]

Identify the cause according to the message, and then remove it.

3.1.31 MCA00031

could not read PID file "%s" of Mirroring Controller detail of cause:"%s"

[Description]

Could not read PID file of Mirroring Controller.

[Parameters]

%s: file name

%s: detail of cause

[Action]

Identify the cause according to the message, and then remove it.

3.1.32 MCA00032

invalid contents of PID file "%s" of Mirroring Controller

[Description]

The contents of PID file of Mirroring Controller is invalid.

[Parameters]

%s: file name

[System Processing]

Processing will be aborted.

[Action]

The following causes could be considered.

- The file was stored or replaced by mistake

- The file was corrupted

When starting Mirroring Controller, move or remove the file shown in the message.

When stopping Mirroring Controller, terminate forcibly mc_keeper process and terminate forcibly mc_agent process with using OS command.

3.1.33 MCA00033

Mirroring Controller is already running

[Description]

Mirroring Controller is already running.

[System Processing]

Processing will be aborted.

[Action]

If needed, stop Mirroring Controller, and re-execute.

3.1.34 MCA00034

cannot execute %s command because Mirroring Controller is not running

[Description]

Cannot execute Mirroring Controller command because Mirroring Controller process is not running.

[Parameters]

%s: command name

[System Processing]

Processing will be aborted.

[Action]

Start Mirroring Controller, and re-execute.

3.1.35 MCA00035

failed to start database instance

[Description]

Failed to start database instance.

[System Processing]

Processing will be aborted.

[Action]

Find the database message output in the log files of database output before this message, and then work around according to the Action of the message.

3.1.36 MCA00036

failed to stop database instance target server:"{0}"

[Description]

Failed to stop database instance.

[Parameters]

{0}: target server ("localhost" or server ID)

[System Processing]

Processing will be aborted.

[Action]

Find the database message output in the log files of database on the target server

output before this message, and then work around according to the Action of the message.

3.1.37 MCA00037

Mirroring Controller option is not installed

[Description]

This functionality is enabled by installing Mirroring Controller option.

[System Processing]

Processing will be aborted.

[Action]

To use this functionality, install Mirroring Controller option,and then re-execute.

3.1.38 MCA00038

starting Mirroring Controller

[Description]

Starting Mirroring Controller.

3.1.39 MCA00039

Mirroring Controller started

[Description]

Mirroring Controller started.

3.1.40 MCA00040

failed to start Mirroring Controller

[Description]

Failed to start Mirroring Controller.

[System Processing]

Processing will be aborted.

[Action]

Work around according to the Action of the message output before this message.

On Windows, if there is no message outputted before this message, please refer to the message outputted to an event log.

3.1.41 MCA00041

stopping Mirroring Controller

[Description]

Stopping Mirroring Controller.

3.1.42 MCA00042

Mirroring Controller stopped target server:"{0}"

[Description]

Mirroring Controller stopped.

[Parameters]

{0}: target server ("localhost" or server ID)

3.1.43 MCA00043

failed to stop Mirroring Controller target server:"{0}"

[Description]

Failed to stop Mirroring Controller.

[Parameters]

{0}: target server ("localhost" or server ID)

[System Processing]

Processing will be aborted.

[Action]

Identify the cause from system log or event log on the target server, and work around.

3.1.44 MCA00044

stopping Mirroring Controller forcibly

[Description]

Stopping Mirroring Controller forcibly.

3.1.45 MCA00045

stopped Mirroring Controller forcibly

[Description]

Mirroring Controller stopped forcibly.

3.1.46 MCA00046

enabled failover target server:"{0}"

[Description]

Enabled failover and automatic detach.

[Parameters]

{0}: target server ("localhost" or server ID)

3.1.47 MCA00047

failed to enable failover target server:"{0}"

[Description]

Failed to enable failover and automatic detach.

[Parameters]

{0}: target server ("localhost" or server ID)

[System Processing]

Processing will be aborted.

[Action]

Identify the cause from messages on system log or event log, and work around.

3.1.48 MCA00048

disabled failover target server:"{0}"

[Description]

Disabled failover and automatic detach.

[Parameters]

{0}: target server ("localhost" or server ID)

3.1.49 MCA00049

failed to disable failover target server:"{0}"

[Description]

Failed to disable failover and automatic detach.

[Parameters]

{0}: target server ("localhost" or server ID)

[System Processing]

Processing will be aborted.

[Action]

Identify the cause from messages on system log or event log, and work around.

3.1.50 MCA00050

{0}: server ID "{2}" specified with option "{1}" does not exist in definition file "{3}"

[Description]

server ID specified with the option does not exist in definition file.

[Parameters]

{0}: command name

{1}: option

{2}: server ID

{3}: file name

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting options or definition file.

3.1.51 MCA00051

{0}: The IP address or host name of the server where the command was executed in is not found on the definition file "{1}"

[Description]

Either of the followings has occurred.

- The IP address or host name that does not exist is specified.

- The network interface is stopped.

[Parameters]

{0}: command name

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter descriptions of "Cluster Operation Guide (Database Multiplexing)".

3.1.52 MCA00052

{0}: wrong server ID "{2}" in definition file "{1}"

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: server ID

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter descriptions of "Cluster Operation Guide (Database Multiplexing)".

3.1.53 MCA00053

failed to detach standby server

[Description]

Failed to detach standby server

because processing of detaching cannot be continued by something failure.

[System Processing]

Processing of detaching will be aborted.

[Action]

Find the message output before this message from display, system log or event log, and then work around according to the Action of the message.

3.1.54 MCA00054

could not write to file "{0}": exception={1}: {2}

[Description]

Failed to detach or synchronize standby server because could not write to the file.

[Parameters]

{0}: file name

{1}: exception type

{2}: exception detail

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes, and detach or synchronize standby server according to "Actions when an Error Occurs" of "Cluster Operation Guide (Database Multiplexing)".

3.1.55 MCA00055

unexpected error occurred in the monitoring process: {0}

[Description]

Monitoring process could not continue because the unexpected error occured.

[Parameters]

{0}: detail of cause

[System Processing]

Continues monitoring.

[Action]

Check the error detail and eliminate causes.

If you cannot clear the problem, contact Fujitsu technical support.

3.1.56 MCA00056

unexpected error occurred: {0}

[Description]

An unexpected error occurred.

[Parameters]

{0}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate cause.

If you cannot clear the problem, contact Fujitsu technical support.

3.1.57 MCA00057

failed to stop Mirroring Controller forcibly

[Description]

Failed to stop Mirroring Controller forcibly.

[System Processing]

Processing will be aborted.

[Action]

Check [Action] of the message output before this message,and re-execute.

If re-execution fails, terminate forcibly mc_keeper process and terminate forcibly mc_agent process with using OS command.

3.1.58 MCA00058

could not access path "%s" specified as a directory for Mirroring Controller detail of cause:"%s"

[Description]

could not access path specified as a directory for Mirroring Controller.

[Parameters]

%s: path name

%s: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.1.59 MCA00059

system call error occurred:"%s" detail of cause:"%s"

[Description]

System call error occurred.

[Parameters]

%s: system call name

%s: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.1.60 MCA00060

could not get installation path

[Description]

Enterprise Postgres may not be installed.

[System Processing]

Processing will be aborted.

[Action]

Re-install Enterprise Postgres.

3.1.61 MCA00061

could not access path "%1$s" for parameter "%2$s" in definition file "%3$s" detail of cause:"%4$s"

[Description]

could not access path for parameter in definition file.

[Parameters]

%3$s: file name

%2$s: parameter name

%1$s: path name

%4$s: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.1.62 MCA00062

promotion processing completed

[Description]

Promotion processing completed.

3.1.63 MCA00063

promotion processing failed

[Description]

Promotion processing failed.

[System Processing]

Processing will be aborted.

[Action]

Clear the problem according to [Action] of the message which was output before this message in system log or in database server log.

3.1.64 MCA00064

stopped database instance forcibly

[Description]

Database instance stopped forcibly.

3.1.65 MCA00065

failed to stop database instance forcibly

[Description]

Failed to stop database instance forcibly.

[System Processing]

Processing will be aborted.

[Action]

Clear the problem according to [Action] of the message which was output before this message in system log or in database server log.

3.1.66 MCA00067

did not switch during a degeneration use

[Description]

Did not switch during a degeneration use.

[System Processing]

Processing will be aborted.

[Action]

If Mirroring Controller executed switching automatically, find the message output before this message from system log or event log to identify the cause of degeneration, and then eliminate causes according to [Action] of the message and try to switch with command.

3.1.67 MCA00068

{0}: users other than an instance administrator have the access privileges for definition file "{1}"

[Description]

users other than an instance administrator have the access privileges for definition file.

[Parameters]

{0}: command name

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Revoke all the access privileges for users other than an instance administrator.

MCA00069

could not execute because Mirroring Controller of the server "{0}" is not running

[Description]

Could not execute because Mirroring Controller is not running.

[Parameters]

{0}: server ID

[System Processing]

Processing will be aborted.

[Action]

Start Mirroring Controller, and try to switch with command.

3.1.68 MCA00070

Try "%s --help" for more information.\n

[Description]

--help option can show more additional information.

[Parameters]

%s: command name

[System Processing]

None.

[Action]

Check the message output before this message, and refer to descriptions shown by '--help' option.

3.1.69 MCA00071

starting to {0}

[Description]

Switching standby server to primary server.

[Parameters]

{0}: "switch over"

3.1.70 MCA00072

failed to {0}

[Description]

Failed to switch to primary server because of an unexpected failure.

[Parameters]

{0}: "switch over"

[System Processing]

Processing of switching will be aborted.

[Action]

Find the message output before this message from display, system log or event log, and then eliminate causes according to [Action] of the message.

3.1.71 MCA00073

error detected in handling of the database instance detail of cause:"{0}"

[Description]

Error detected in handling of the database instance for the following purposes.

- Obtain the port number of database instance

- Access to the data storage destination directory

[Parameters]

{0}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.1.72 MCA00074

could not read PID file "{0}" of Mirroring Controller detail of cause:"{1}"

[Description]

Could not read PID file of Mirroring Controller.

[Parameters]

{0}: file name

{1}: detail of cause

[Action]

Identify the cause according to the message, and then remove it.

3.1.73 MCA00075

invalid contents of PID file "{0}" of Mirroring Controller

[Description]

The contents of PID file of Mirroring Controller is invalid.

[Parameters]

{0}: file name

[System Processing]

Processing will be aborted.

[Action]

The following causes could be considered.

- The file was stored or replaced by mistake

- The file was corrupted

When starting Mirroring Controller, move or remove the file shown in the message.

When stopping Mirroring Controller, terminate forcibly mc_keeper process and terminate forcibly mc_agent process with using OS command.

3.1.74 MCA00076

cannot execute "{0}" command because Mirroring Controller is not running

[Description]

Cannot execute Mirroring Controller command because Mirroring Controller process is not running.

[Parameters]

{0}: command name

[System Processing]

Processing will be aborted.

[Action]

Start Mirroring Controller, and re-execute.

3.1.75 MCA00077

%s: argument of option "%s" is too long

[Description]

Argument of option is too long.

[Parameters]

%s: command name

%s: option

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting options.

3.1.76 MCA00078

%s: invalid option -- %s

[Description]

Invalid option.

[Parameters]

%s: command name

%s: option

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting options.

3.1.77 MCA00079

%s: unnecessary operand "%s"

[Description]

Unnecessary operand.

[Parameters]

%s: command name

%s: operand

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting operand.

3.1.78 MCA00080

%s: unrecognized operation mode or no operation mode specified

[Description]

Unrecognized operation mode or no operation mode specified.

[Parameters]

%s: command name

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting or specifying operation mode.

3.1.79 MCA00081

start to enable the parameter "{1}" required to build in the standby server "{0}"

[Description]

Start to enable the parameter required to build in the standby server.

[Parameters]

{0}: server ID

{1}: parameter name

3.1.80 MCA00082

enableing the parameter "{1}" required to build in the standby server "{0}" completed

[Description]

Enableing the parameter required to build in the standby server completed.

[Parameters]

{0}: server ID

{1}: parameter name

3.1.81 MCA00083

failed to enable the parameter "{1}" required to build in the standby server "{0}"

[Description]

Failed to enable the parameter required to build in the standby server.

The following causes could be considered.

- another command is running

- can not access definition file

- parameter does not exist

[Parameters]

{0}: server ID

{1}: parameter name

[System Processing]

Processing will be aborted.

[Action]

- If the parameter is not set

On the primary server, set the parameter of postgresql.conf file according to "parameter" description of "Cluster Operation Guide (Database Multiplexing)" and execute pg_ctl command with reload mode.

- Otherwise

Find the message output before this message from display, system log or event log, and then eliminate causes according to [Action] of the message. Then, on the primary server, set the parameter of postgresql.conf file according to "parameter" description of "Cluster Operation Guide (Database Multiplexing)" and execute pg_ctl command with reload mode.

3.1.82 MCA00084

primary server is already running

[Description]

Primary server is already running.

[System Processing]

Processing will be aborted.

[Action]

The standby server might be running without creating standby.signal. Create standby.signal, and re-execute.

3.1.83 MCA00085

cannot start Mirroring Controller because database instance is not running

[Description]

Cannot start Mirroring Controller because database instance is not running.

[System Processing]

Processing will be aborted.

[Action]

Start database instance, and re-execute.

3.1.84 MCA00086

could not get state of database instance detail of cause:"{0}"

[Description]

Could not get state of database instance.

[Parameters]

{0}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.1.85 MCA00087

unusable character is included in path "%s" specified as a directory for Mirroring Controller

[Description]

Unusable character is included in path specified as a directory for Mirroring Controller.

[Parameters]

%s: path name

[System Processing]

Processing will be aborted.

[Action]

Correct the path specified as a directory for Mirroring Controller according to the message and mc_ctl command descriptions of "Reference".

3.1.86 MCA00088

%1$s: unusable character is included in server ID "%3$s" specified with option "%2$s"

[Description]

Unusable character is included in server ID specified with option.

[Parameters]

%1$s: command name

%2$s: option

%3$s: server ID

[System Processing]

Processing will be aborted.

[Action]

Correct the server ID specified with option according to the message and mc_ctl command descriptions of "Reference".

3.1.87 MCA00089

only instance administrator can execute this command

[Description]

Only instance administrator who created the directory for Mirroring Controller can execute this command.

[System Processing]

Processing will be aborted.

[Action]

Re-execute the command by the instance administrator who created the directory for Mirroring Controller.

3.1.88 MCA00090

could not read file "{0}": Permission denied

[Description]

No read permissions for the file.

[Parameters]

{0}: file name

[System Processing]

Processing will be aborted.

[Action]

Re-execute the command, after granting the read permissions to the file.

3.1.89 MCA00091

host name or IP address "{1}" of the primary server and the standby server in definition file "{0}" are same, but the --local-server option was not specified

[Description]

Host name or IP address of the primary server and the standby server in definition file are same, but the --local-server option was not specified.

[Parameters]

{0}: file name

{1}: host name or IP address

[System Processing]

Processing will be aborted.

[Action]

If the primary server and the standby server are built in the same server, execute the mc_ctl command with the --local-server option.

If the primary server and the standby server are built in the different server, correct host name or IP address in the definition file.

3.1.90 MCA00092

this feature is not available in this edition

[Description]

This feature is not available in this edition.

[System Processing]

Processing will be aborted

[Action]

Please install the right edition for this feature.

3.1.91 MCA00093

installation environment is destroyed

[Description]

Enterprise Postgres may not be installed correctly or may be destroyed.

[System Processing]

Processing will be aborted.

[Action]

Re-install Enterprise Postgres.

3.1.92 MCA00094

%s: invalid argument for option %s

[Description]

Invalid argument for option.

[Parameters]

%s: command name

%s: option

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting options.

3.1.93 MCA00095

could not remove file or directory "{0}"

[Description]

Could not remove the file or the directory.

[Parameters]

{0}: file name or directory name

[System Processing]

Processing will be aborted.

[Action]

Check the status of the file or the directory and eliminate causes, and then remove it.

3.1.94 MCA00096

could not write file "{0}": exception={1}: {2}

[Description]

Could not write the file.

[Parameters]

{0}: file name

{1}: exception type

{2}: exception detail

[System Processing]

Processing will be aborted.

[Action]

Identify the cause according to the message, and then remove it.

3.1.95 MCA00097

setup of standby server completed

[Description]

Setup of standby server completed.

3.1.96 MCA00098

setup of standby server failed

[Description]

Setup of standby server failed.

[System Processing]

Processing will be aborted.

[Action]

Find the message output before this message from display, system log or event log, and then eliminate causes according to [Action] of the message.

3.1.97 MCA00099

{0}: server""{1}"" is running as a standby server

[Description]

Could not continue processing because database instance to be duplicated is not running as a primary server.

[Parameters]

{0}: command name

{1}: server ID

[System Processing]

Processing will be aborted.

[Action]

Re-execute this command on the standby server to be set up.

3.2 Message Numbers Beginning with MCA00100

3.2.1 MCA00100

cannot execute {0} command because Mirroring Controller is running

[Description]

Cannot execute this command because Mirroring Controller is running on the server to be set up.

[Parameters]

{0}: command name

[System Processing]

Processing will be aborted.

[Action]

Execute this command on server where primary server is not running. If execute it on the right server, stop Mirroring Controller and then re-execute it.

3.2.2 MCA00101

cannot execute {0} command because database instance is running

[Description]

Cannot execute this command because database instance is running on the server to be set up.

[Parameters]

{0}: command name

[System Processing]

Processing will be aborted.

[Action]

Execute this command on server where primary server is not running. If execute it on the right server, stop database instance and then re-execute it.

3.2.3 MCA00102

{0}: invalid argument value {2} for option {1}

[Description]

Invalid argument for option.

[Parameters]

{0}: command name

{1}: option

{2}: argument value

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting argument value for the option.

3.2.4 MCA00103

could not move file or directory from {0} to {1}

[Description]

Could not move the file or the directory.

[Parameters]

{0}: source file or directory

{1}: target file or directory

[System Processing]

Processing will be aborted.

[Action]

Check the status of the file or the directory and eliminate causes, and then remove it.

3.2.5 MCA00104

could not create directory {0}

[Description]

Could not create the directory.

[Parameters]

{0}: target directory

[System Processing]

Processing will be aborted.

[Action]

Check the status of the directory and eliminate causes, and then remove it.

3.2.6 MCA00105

could not read the access privileges of {0}

[Description]

Could not read the access privileges.

[Parameters]

{0}: target directory

[System Processing]

Processing will be aborted.

[Action]

Check the status of the directory and eliminate causes, and then remove it.

3.2.7 MCA00106

failed to set the access privileges of {0}

[Description]

Failed to set the access privileges.

[Parameters]

{0}: target directory

[System Processing]

Processing will be aborted.

[Action]

Check the status of the directory and eliminate causes, and then remove it.

3.2.8 MCA00107

service "{0}" is not registered

[Description]

Service is not registered.

[Parameters]

{0}: Service name

[System Processing]

Processing will be aborted.

[Action]

Register service, and re-execute.

3.2.9 MCA00108

could not start service "{0}" detail of cause:"{1}"

[Description]

Could not start service.

[Parameters]

{0}: Service name

{1}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.2.10 MCA00109

could not start service "%s" detail of cause:"%s"

[Description]

Could not start service.

[Parameters]

%s: Service name

%s: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.2.11 MCA00110

service "%s" is not registered

[Description]

Service is not registered.

[Parameters]

%s: Service name

[System Processing]

Processing will be aborted.

[Action]

Register service, and re-execute.

3.2.12 MCA00111

Mirroring Controller service "%s" has been registered

[Description]

Mirroring Controller service has been registered with Windows Service.

[Parameters]

%s: Service name

3.2.13 MCA00112

Mirroring Controller service "%s" has been unregistered

[Description]

Mirroring Controller service has been unregistered from Windows Service.

[Parameters]

%s: Service name

3.2.14 MCA00113

service name "%s" is already in use

[Description]

Service name is already in use.

[Parameters]

%s: Service name

[System Processing]

Processing will be aborted.

[Action]

Check the service name, and re-execute.

3.2.15 MCA00114

could not register service "%s" detail of cause: "%s"

[Description]

An error occurred during registration of service.

[Parameters]

%s: Service name

%s: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.2.16 MCA00115

could not unregister service "%s" detail of cause: "%s"

[Description]

An error occurred during deregistration of service.

[Parameters]

%s: Service name

%s: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.2.17 MCA00116

%s: option "%s" is required

[Description]

A required option is not specified.

[Parameters]

%s: command name

%s: option

[System Processing]

Processing will be aborted.

[Action]

Specify the required option, and re-execute.

3.2.18 MCA00117

no authority to execute this command

[Description]

Only the administrative user can run this command.

[System Processing]

Processing will be aborted.

[Action]

Invoke the administrator's prompt, and re-execute this command.

3.2.19 MCA00119

could not receive respons from {0} server({1})

[Description]

An error was detected on the server.

[Parameters]

{0}: monitored object (server)

{1}: server type (primary, candidate primary, standby)

[System Processing]

Perform failover or detaching.

If failover or detaching is already performed, these functions would be disabled.

[Action]

Refer to the description about workaround for failure of "Cluster Operation Guide (Database Multiplexing)".

3.2.20 MCA00120

detected streaming replication error in {0} server({1})

[Description]

A streaming replication error was detected.

[Parameters]

{0}: monitored object (database process)

{1}: server type (primary, candidate primary, standby)

[System Processing]

Perform failover or detaching.

If failover or detaching is already performed, these functions would be disabled.

[Action]

Refer to the description about workaround for failure of "Cluster Operation Guide (Database Multiplexing)".

3.2.21 MCA00121

disk drives are available

[Description]

Database where data storage, transaction log storage and tablespaces are saved is working correctly.

3.2.22 MCA00122

a {0} server({1}) is running normally

[Description]

A server is running normally.

[Parameters]

{0}: server type (primary, candidate primary, standby)

{1}: monitored object (server)

3.2.23 MCA00123

Streaming Replication has started

[Description]

Streaming Replication has started

3.2.24 MCA00124

postmaster is running in {0} server({1})

[Description]

postmaster is running

[Parameters]

{0}: server type (primary, candidate primary, standby)

{1}: monitored object (server)

3.2.25 MCA00125

failed to get the standby server information

[Description]

Failed to get the standby server information

[System Processing]

Perform failover or detaching.

If failover or detaching is already performed, these functions would be disabled.

[Action]

Refer to the description about workaround for failure of "Cluster Operation Guide (Database Multiplexing)".

3.2.26 MCA00126

{0} server({1}) was downed

[Description]

database server was downed

[Parameters]

{0}: server type (primary, candidate primary, standby)

{1}: monitored object (server)

[System Processing]

Perform failover or detaching.

If failover or detaching is already performed, these functions would be disabled.

[Action]

Refer to the description about workaround for failure of "Cluster Operation Guide (Database Multiplexing)".

3.2.27 MCA00127

You can promote the standby server

[Description]

You can promote the standby server

3.2.28 MCA00128

You cannot promote the standby server

[Description]

You cannot promote the standby server

3.2.29 MCA00129

detected a disk I/O error in {0} server({1})

[Description]

A disk I/O error was detected.

[Parameters]

{0}: server type (primary, candidate primary, standby)

{1}: monitored object (server)

[System Processing]

Perform failover or detaching.

If failover or detaching is already performed, these functions would be disabled.

[Action]

Refer to the description about workaround for failure of "Cluster Operation Guide (Database Multiplexing)".

3.2.30 MCA00130

starting to switch over forcibly

[Description]

Starting to switch over forcibly.

3.2.31 MCA00131

succeeded in switching over to {0} forcibly

[Description]

Switching standby server to primary server succeeded.

[Parameters]

{0}: server ID of standby server

3.2.32 MCA00132

failed to switch over to {0} forcibly

[Description]

Failed in processing of switching because of unexpected failure.

[Parameters]

{0}: server ID of standby server

[System Processing]

Processing of switching will be aborted.

[Action]

Find the message output before this message from display, system log or event log, and then work around according to the Action of the message.

3.2.33 MCA00133

starting to detach standby server forcibly

[Description]

Starting to detach standby server forcibly.

3.2.34 MCA00134

succeeded in detaching standby server "{0}" completed forcibly

[Description]

Detached standby server succeeded.

[Parameters]

{0}: server ID

3.2.35 MCA00135

failed to detach standby server "{0}" forcibly

[Description]

Failed to detach standby server.

[Parameters]

{0}: server ID

[System Processing]

Processing of detaching will be aborted.

[Action]

Find the message output before this message from display, system log or event log, and then work around according to the Action of the message.

3.2.36 MCA00136

{0}: the specified option is invalid: "{1}"

[Description]

the specified option is invalid.

[Parameters]

{0}: command name

{1}: option name

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting options.

3.2.37 MCA00137

disabled standby server "{0}" to automatically switch because of canceling the synchronous replication for standby server

[Description]

Disabled the standby server to automatically switch because of canceling the synchronous replication for standby server.

[Parameters]

{0}: server ID

3.2.38 MCA00138

disabled standby server "{0}" to automatically switch

[Description]

Disabled the standby server to automatically switch.

[Parameters]

{0}: server ID

3.2.39 MCA00139

failed to disable standby server "{0}" to automatically switch

[Description]

Failed to disable the standby server to automatically switch.

[Parameters]

{0}: server ID

[System processing]

Processing will be aborted.

[Action]

The following causes could be considered.

- An error was detected in database server

- Timeout waiting for between Mirroring Controller processes of database server

Identify the cause from messages on system log or event log, and work around.

3.2.40 MCA00140

{0}: invalid server kind specified in definition file "{1}" line {2}

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: line number

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.2.41 MCA00141

{0}: server kind 'arbiter' do not specified in definition file "{2}" despite value 'arbitration' is specified parameter heartbeat_error_action in definition file "{1}"

[Description]

Server kind 'arbiter' do not been specified in network.conf despite value 'arbitration' is specified parameter heartbeat_error_action in "server identifer".conf

[Parameters]

{0}: command name

{1}: file name

{2}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.2.42 MCA00142

{0}: could not specify server kind 'arbiter' in definition file "{3}" because value '{2}' is specified parameter heartbeat_error_action in definition file "{1}"

[Description]

Could not specify server kind 'arbiter' in network.conf because value 'arbitration' is not specified parameter heartbeat_error_action in "server identifer".conf.

[Parameters]

{0}: command name

{1}: file name

{2}: message, arbitration, command, fallback

{3}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.2.43 MCA00143

{0}: second port number to use for the arbitration network is not specified in definition file "{1}"

[Description]

Second port number to use for the arbitration network is not specified in network.conf.

[Parameters]

{0}: command name

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.2.44 MCA00144

{0}: could not specify second port number to use for the arbitration network in definition file "{3}" because value '{2}' is specified parameter heartbeat_error_action in definition file "{1}"

[Description]

Could not specify second port number to use for the arbitration network in network.conf.

[Parameters]

{0}: command name

{1}: file name

{2}: message, command, fallback

{3}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.2.45 MCA00145

{0}: second IP address to use for the arbitration network is not specified in definition file "{1}"

[Description]

Second IP address to use for the arbitration network is not specified in network.conf.

[Parameters]

{0}: command name

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.2.46 MCA00146

{0}: cannot specify second IP Address on server kind 'arbiter' in definition file "{1}"

[Description]

Cannot specify second IP Address on server kind 'arbiter' in network.conf.

[Parameters]

{0}: command name

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.2.47 MCA00147

{0}: could not use parameter '{3}' because value '{2}' is specified parameter heartbeat_error_action in definition file "{1}"
[Discription]
Could not use parameter in "server identifier".conf.

[Parameters]

{0}: command name

{1}: file name

{2}: message, arbitration, command, fallback

{3}: parameter

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.2.48 MCA00148

{0}: could not use option "{3}" because value '{2}' is specified parameter heartbeat_error_action in definition file "{1}"

[Description]

Could not use this option.

[Parameters]

{0}: command

{1}: file name

{2}: message, arbitration, command, fallback

{3}: option

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting options according to the message and mc_ctl command descriptions of "Reference".

3.2.49 MCA00149

requesting arbitration server "{0}" to connect

[Description]

Requesting arbitration server to connect.

[Parameters]

{0}: server ID

[System processing]

Requesting arbitration server to connect.

3.2.50 MCA00150

trying to connect to arbitration server "{0}"

[Description]

Trying to connect to arbitration server.

[Parameters]

{0}: server ID

[System processing]

Trying to connect to arbitration server until success.

3.2.51 MCA00151

succeeded in connection with arbitration server "{0}"

[Description]

Succeeded in connection with arbitration server.

[Parameters]

{0}: server ID

3.2.52 MCA00152

failed to connect to arbitration server "{0}" event:"{1}"

[Description]

Either of the followings has occurred.

- incorrect specification in network.conf

- error occurs in the network between database server and arbitration server

- Mirroring Controller Arbitration process is not running or in the stop processing

- Mirroring Controller Arbitration process or arbitration server detects an error

[Parameters]

{0}: server ID

{1}: "timeout" or "communication error"

[System Processing]

Processing will be aborted.

[Action]

Check the following and identify the cause, and eliminate cause.

- specification about server kind 'arbiter' in network.conf

- the value of arbiter_connect_timeout parameter in "server identifier".conf

- communication status between database server and arbitration server

- Mirroring Controller Arbitration process staring status

- the message in arbitration server

3.2.53 MCA00153

disconnected from arbitration server "{0}"

[Description]

Disconnected from arbitration server.

[Parameters]

{0}: sever ID

3.2.54 MCA00154

timeout waiting for communication with Mirroring Controller Arbitration process server:"{0}"

[Description]

Timeout waiting for communication between Mirroring Controller process and Mirroring Controller Arbitration process.

[Parameters]

{0}: server ID

[System Processing]

Try to connect to arbitration server.

[Action]

Check whether a network error between arbitration server and database server or an error of arbtration server was detected.

If an error was not detected, the value of arbiter_alive_timeout parameter in "server identifier".conf is too short.

Review and extend the value of arbiter_alive_timeout in "server identifier".conf.

3.2.55 MCA00155

failed to open communication environment detail of cause:"{0}"

[Description]

Failed to open communication environment.

[Parameters]

{0}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

3.2.56 MCA00156

invalid port number for server "{0}" in definition file "{1}"

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: server ID

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.2.57 MCA00157

communication error with the arbitration server "{0}" occurred

[Description]

Communication error with the arbitration server occurred.

[Parameters]

{0}: server ID

[System Processing]

Trying to connect.

[Action]

Either of the followings has occurred.

- When Mirroring Controller Arbitration process of a arbitration server is not running

- When abnormality occurs in the network between a database server and arbitration server

Identify the cause from messages on system log or event log in arbitration server or database server, and work around.

3.2.58 MCA00158

failed to start Mirroring Controller because of could not connect to arbitration server "{0}"

[Description]

Failed to start Mirroring Controller because of could not connect to arbitration server.

[Parameters]

{0}: server ID

[System processing]

Processing will be aborted.

[Action]

Work around according to the Action of the message output before this message.

On Windows, if there is no message outputted before this message, please refer to the message outputted to an event log.

MCA00159

requesting arbitration server "{1}" to arbitrate for database server "{0}"

[Description]

Requesting arbitration server to arbitrate for the target server.

[Parameters]

{0}: server ID

{1}: server ID

[System processing]

Requesting arbitration server to arbitrate for the target server.

3.2.59 MCA00160

the arbitration server "{0}" arbitrated status of database server "{1}" as sanity

[Description]

The arbitration server arbitrated status of database server as sanity.

[Parameters]

{0}: server ID

{1}: server ID

3.2.60 MCA00161

the request for arbitration was omitted because fencing command for a database server "{0}" was finished already

[Description]

The request for arbitration was omitted because fencing command for a database server was finished already.

[Parameters]

{0}: server ID

3.2.61 MCA00162

arbitration processing cannot request because the "{0}" command is carrying out

[Description]

Cannot request arbitration processing because command is running on the same or another server.

[Parameters]

{0}: command name

[System Processing]

Processing will be aborted.

3.2.62 MCA00163

timeout waiting for requesting arbitration server "{0}" to arbitrate

[Description]

Timeout waiting for requesting arbitration server to arbitrate.

[Parameters]

{0}: server ID

[System Processing]

Processing will be aborted.

[Action]

Switch or detach server in yourself according to descriptions of "Cluster Operation Guide (Database Multiplexing)".

3.2.63 MCA00164

requesting arbitration server "{0}" to fence database server "{1}"

[Description]

Requesting arbitration server to fence database server.

[Parameters]

{0}: server ID

{1}: server ID

[System Processing]

Requesting arbitration server to fence.

3.2.64 MCA00165

arbitration server "{0}" succeeded in fencing for database server "{1}"

[Description]

Arbtration server succeeded in executing fencing command for the target server.

[Parameters]

{0}: server ID

{1}: server ID

3.2.65 MCA00166

arbitration server "{0}" failed to fence for database server "{1}". detail of cause: check log file in arbitration server

[Description]

Arbitration process failed to execute fencing command for the target server.

[Parameters]

{0}: server ID

{1}: server ID

[System Processing]

Processing will be aborted.

[Action]

Check error message in arbitration server.

If needed, perform switch or detach the database server with manually operation according to descriptions of "Cluster Operation Guide (Database Multiplexing)".

3.2.66 MCA00167

timeout waiting for a request to fence to arbitration server "{0}"

[Description]

Timeout waiting for a request to fence to arbitration server.

[Parameters]

{0}: server ID

[System Processing]

Processing will be aborted.

[Action]

Switch or detach server in yourself according to descriptions of "Cluster Operation Guide (Database Multiplexing)".

3.2.67 MCA00168

arbitration server "{0}" rejected the request to fence database server "{1}"

[Description]

Either of the followings is esteemed.

- The arbtration server is executing arbitration process

- Fencing command was executed just before

[Parameters]

{0}: server ID

{1}: server ID

3.2.68 MCA00169

could not request "{1}" because of disconnecting from arbitration server "{0}"

[Description]

Could not request arbitration server to arbitrate, fence or disable automatically switch because of disconnecting.

[Parameters]

{0}: server ID

{1}: "arbitration" or "fencing" or "disable automatically switch"

[System Processing]

Processing will be aborted.

[Action]

Either of the followings has occurred.

- Mirroring Controller Arbitration process is not running in arbitration server

- Abnormality occurs in the network between database server and arbitration server

Identify the cause from messages on system log or event log in arbitration server or database server, and work around.

3.2.69 MCA00170

rejeceted a request arbitration server "{1}" because database server "{0}" was beening fencing

[Description]

The following requirements shouldn't be executed if it is requested from the database server which is a fencing target.

- request to arbitrate

- request to fence

[Parameters]

{0}: server ID

{1}: server ID

[Action]

Processing will be aborted.

3.2.70 MCA00171

requested arbitration server "{0}" to disable standby server "{1}" to automatically switch

[Description]

Requested arbitraion server to disable the standby server to automatically switch.

[Parameters]

{0}: server ID

{1}: server ID

3.2.71 MCA00172

arbitration server "{0}" fenced standby server "{1}" because of failed to disable the standby server to automatically switch

[Description]

Arbitration server fenced the standby server because of failed to disable the standby server to automatically switch.

[Parameters]

{0}: server ID

{1}: server ID

3.2.72 MCA00173

arbitration server "{0}" tried to fence standby server "{1}" because of failed to disable the standby server to automatically switch, however fencing was failed

[Description]

Arbitration server tried to fence the standby server because of failed to disable the standby server to automatically switch, however fencing was failed.

[Parameters]

{0}: server ID

{1}: server ID

[System Processing]

Processing will be aborted.

[Action]

Switch or detach database server which error has been detected in yourself.

After, identify the cause from messages on system log or event log in arbitration server and work around.

3.2.73 MCA00174

requested standby server "{0}" to disable automatically switching

[Description]

Requested the standby server to disable automatically switching.

[Parameters]

{0}: server ID

3.2.74 MCA00175

{0}: cannot execute switching over forcibly other than on standby server

[Description]

Switching over forcibly needs to be executed only on standby server.

[Parameters]

{0}: command name

[System Processing]

Processing will be aborted.

[Action]

Check whether the server where the command was executed is correct.

Either, check that Mirroring Controller has been degenerate state yet.

3.2.75 MCA00176

{0}: cannot execute detaching forcibly other than on primary server

[Description]

Detaching forcibly needs to be executed only on pimary server.

[Parameters]

{0}: command name

[System Processing]

Processing will be aborted.

[Action]

Check whether the server where the command was executed is correct.

Either, check that Mirroring Controller has been degenerate state yet.

3.2.76 MCA00177

forcible switch over to standby server "{0}" was requested although the data may be not synchronous with primary server

[Description]

Forcible switch over to standby server was requested although the data may be not synchronous with primary server.

[Parameters]

{0}: server ID

[System Processing]

Processing will be continued.

[Action]

Check the data of database server and recovery as necessary, after forcible switch over will be completed.

3.2.77 MCA00178

database server which is able to be switched over is not found

[Description]

Database server which is able to be switched over is not found.

[System Processing]

Processing will be aborted.

3.2.78 MCA00179

starting to detach standby server

[Description]

Starting to detach standby server.

3.2.79 MCA00180

database server which needs to be detached is not found

[Description]

Database server which needs to be detached is not found.

[System Processing]

Processing will be aborted.

3.2.80 MCA00181

database server already has been detached

[Description]

Database server already has been detached.

[System Processing]

Processing will be aborted.

3.2.81 MCA00182

failed to switch over forcibly

[Description]

Failed in processing of switching because of unexpected failure.

[System Processing]

Processing of switching will be aborted.

[Action]

Find the message output before this message from display, system log or event log, and then work around according to the Action of the message.

3.2.82 MCA00183

failed to detach standby server forcibly

[Description]

Failed to detach standby server.

[System Processing]

Processing of detaching will be aborted.

[Action]

Find the message output before this message from display, system log or event log, and then work around according to the Action of the message.

3.2.83 MCA00184

could not {1} because {0} server is abnormal

[Description]

Could not switch over or detach because server is abnormal

[Parameters]

{0}: "primary" or "candidate primary" or "standby"

{1}: "switch over" or "detach"

[System Processing]

Processing will be aborted.

[Action]

Find the message output before this message in the log files on the target server, and work around according to the Action of the message.

3.2.84 MCA00185

value of heartbeat_error_action is different from the value of other server "{0}"

[Description]

Value of heartbeat_error_action is different from the value of other server "{0}".

[Parameters]

{0}: server ID

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting value of heartbeat_error_action to the same value as another server.

3.2.85 MCA00186

restarting Mirroring Controller process because an its error was detected: %s

[Description]

Restarting Mirroring Controller process because an its error was detected.

[Parameters]

%s: error detail ("no response", "down")

[System Processing]

Restarting Mirroring Controller process.

3.2.86 MCA00187

Mirroring Controller process was restarted

[Description]

Mirroring Controller process was restarted.

3.2.87 MCA00188

failed to restart Mirroring Controller

[Description]

Failed to restart Mirooring Controller.

[System Processing]

Processing will be aborted.

[Action]

Work around according to the Action of the message output before this message.

On Windows, if there is no message outputted before this message, please refer to the message outputted to an event log.

After, execute the mc_ctl command to restart Mirroring Controller.

3.2.88 MCA00189

could not access "{0}" file detail of cause:"{1}"

[Description]

Could not access the process information file of OS.

[Parameters]

{0}: file name

{1}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.2.89 MCA00190

invalid contents of "{0}" file

[Description]

Invalid contents of /proc/[pid]/status file.

[Parameters]

{0}: file name

[System Processing]

Processing will be aborted.

[Action]

Check whether mc_keeper process or mc_agent process exists with using OS command.

3.2.90 MCA00191

executing arbitration command and inquiring the result of arbitration

[Description]

Executing arbitration command that execute arbitration.

[System processing]

Degenerate depending on result of arbitration command.

3.2.91 MCA00192

executing fencing command

[Description]

Executing fencing command.

[System processing]

Processing will be continued depending on result of fencing command.

3.2.92 MCA00193

starting degenerate because of result of arbitration command for database server "{0}" result:"{1}"

[Description]

Starting degenerate because of result of arbitration command for target server.

[parameters]

{0}: server ID

{1}: return code of command

3.2.93 MCA00194

degeneracy is not executed because of result of arbitration command to database server "{0}" result:"{1}"

[Description]

Degeneracy is not executed because of result of arbitration command to target server.

[parameters]

{0}: server ID

{1}: return code of command

3.2.94 MCA00195

timeout has occurred during arbitration command has been executing to database server "{0}"

[Description]

Timeout has occurred during arbitration command has been executing to target server.

[Parameters]

{0}: server ID

[System Processing]

Processing will be aborted.

[Action]

Check the state of database server and switch or detach the server with manually operation according to descriptions of "Cluster Operation Guide (Database Multiplexing)" as necessary.

When find process ID of arbitration command, terminate forcibly by using OS command.

3.2.95 MCA00196

{0}: two or more standby server names are specified in the parameter "synchronous_standby_names" in PostgreSQL

[Description]

multiple synchronous standby servers can not be monitored by Mirroring Controller.

[Parameters]

{0}: command name

[System Processing]

Processing will be aborted.

[Action]

Re-execute after specifying single standby server to synchronous_standby_names.

3.2.96 MCA00197

the arbitration server "{0}" didn't arbitrate status of database server "{1}" as sanity

[Description]

Status of the target server was not arbitrated as sanity by arbitration server.

[Parameters]

{0}: server ID

{1}: server ID

[System processing]

Start degenerate processing.

3.2.97 MCA00198

fencing command for database server "{0}" succeeded: result:"{1}"

[Description]

Fencing command for database server succeeded.

[parameters]

{0}: server ID

{1}: return code of command

[System processing]

Processing will be continued.

3.2.98 MCA00199

fencing command for database server "{0}" failed: result:"{1}"

[Description]

Fencing command for database server failed.

[parameters]

{0}: server ID

{1}: return code of command

[System processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

Switch or detach the server with manually operation according to descriptions of "Cluster Operation Guide (Database Multiplexing)" as necessary.

MCA00200

timeout waiting for the fencing command

[Description]

Timeout waiting for the fencing command.

[System processing]

Processing will be aborted.

[Action]

The value of fencing_command_timeout parameter in server identifer.conf is too short.

Review and extend the value of fencing_command_timeout parameter in arbitration.conf.

When find process ID of fencing command, terminate forcibly by using OS command.

3.3 Message Numbers Beginning with MCA00200

3.3.1 MCA00201

executing state-transition-command kind of command:{0}

[Description]

Executing state-transition-command.

[parameters]

{0}: kind of state-transition-command(post-switch, pre-detach, post-attach)

3.3.2 MCA00202

state-transition-command has done kind of command:{0}

[Description]

State-transition-command has done.

[parameters]

{0}: kind of state-transition-command(post-promote, pre-detach, post-attach)

[System processing]

Processing will be continued.

3.3.3 MCA00203

timeout has occurred to state-transition-command kind of command:{0}

[Description]

Timeout has occurred to state-transition-command.

[parameters]

{0}: kind of state-transition-command(post-promote, pre-detach, post-attach)

[System Processing]

Processing of state-transition-command will be aborted.

[Action]

If the process of state-transition-command remained, terminate the process forcibly by using OS command. After checking processing status of state-transition-command, exec the processing of the command manually.

3.3.4 MCA00204

detected recovery from an error on monitoring of table space({0})

[Description]

detected recovery from an error on monitoring of table space

[Parameters]

{0}: directory path of table space

3.3.5 MCA00205

detected no responce on monitoring of database process

[Description]

Detected no responce on monitoring of database process.

[System Processing]

Continues processing.

[Action]

Check the error detail and eliminate causes.

3.3.6 MCA00206

detected recovery from no responce on monitoring of database process

[Description]

Detected recovery from no responce on monitoring of database process.

3.3.7 MCA00207

{0}: the user name specified in parameter db_instance_username of definition file "{1}" is not database superuser

[Description]

The user name specified in parameter db_instance_username is not database superuser.

[Parameters]

{0}: command name

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Re-execute after specifying a database superuser.

3.3.8 MCA00208

{0}: the user name specified in parameter db_instance_username of definition file "{1}" does not exist

[Description]

The user name specified in parameter db_instance_username does not exist.

[Parameters]

{0}: command name

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Re-execute after specifying a database superuser.

3.3.9 MCA00209

{0}: could not specify second IP address to use for the arbitration network in definition file "{3}" because value '{2}' is specified parameter heartbeat_error_action in definition file "{1}"

[Description]

Could not specify second IP address to use for the arbitration network in network.conf.

[Parameters]

{0}: command name

{1}: file name

{2}: message, command, fallback

{3}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.3.10 MCA00210

{0}: cannot specify second port number on server kind 'arbiter' in definition file "{1}"

[Description]

Cannot specify second port number on server kind 'arbiter' in network.conf.

[Parameters]

{0}: command name

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.3.11 MCA00211

{0}: could not use parameter "{2}" because parameter "{3}" is not specified in definition file "{1}"

[Description]

This parameter cannot be specified because the parameter with dependency is not specified in the definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: parameter name

{3}: parameter name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.3.12 MCA00212

{0}: primary_conninfo parameter is not specified in postgresql.auto.conf

[Description]

Primary_conninfo parameter is not specified in postgresql.auto.conf file.

[Parameters]

{0}: command name

[System Processing]

Processing will be aborted.

[Action]

Specify primary_conninfo parameter in postgresql.auto.conf file according to the description of "Setting Up the Standby Center" for disaster recovery of "Cluster Operation Guide (Database Multiplexing)".

3.3.13 MCA00213

values of parameters for abnormality monitoring of operating system or server in server definition file "{1}" are too small for value of heartbeat_interval in arbitration definition file of arbitration server "{0}"

[Description]

Because the values of parameters for abnormality monitoring of the operating system or server in the server definition file are too small compared with the value of heartbeat_interval in the arbitration definition file of the arbitration server, the arbtration for the target database server might be delayed.

[Parameters]

{0}: server ID

{1}: file name

[System Processing]

Depending on the start mode of the mc_ctl command, either of the following processes will be performed.

- When --async-connect-arbiter option is not specified

Processing will be aborted.

- When --async-connect-arbiter option is specified

Continues processing.

[Action]

Take either of the following actions.

- When --async-connect-arbiter option is not specified

Correct the value of parameters for abnormality monitoring of the operating system or server according to the message and "Tuning for Optimization of Degradation Using Abnormality Monitoring With the Arbitration Server" of "Cluster Operation Guide (Database Multiplexing)".

After that, re-execute the mc_ctl command.

- When --async-connect-arbiter option is specified

Correct the parameters for abnormality monitoring of the operating system or server according to the message and "Tuning for Optimization of Degradation Using Abnormality Monitoring With the Arbitration Server" of "Cluster Operation Guide (Database Multiplexing)".

After that, execute the mc_ctl command to restart Mirroring Controller.

3.3.14 MCA00214

{0}: keyword "{1}" is not specified for primary_conninfo parameter in postgresql.auto.conf

[Description]

The required keyword is not specified for primary_conninfo parameter in postgresql.auto.conf file.

[Parameters]

{0}: command name

{1}: keyword name

[System Processing]

Processing will be aborted.

[Action]

Specify the required keyword for primary_conninfo parameter in postgresql.auto.conf file according to the description of "Setting Up the Standby Center" for disaster recovery of "Cluster Operation Guide (Database Multiplexing)".

3.3.15 MCA00215

{0}: keyword "{3}" is not specified for parameter "{2}" in definition file "{1}"

[Description]

The keyword is not specified for the parameter in the definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: parameter name

{3}: keyword name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.3.16 MCA00216

{0}: value that cannot be specified for keyword "{3}" of parameter "{2}" in definition file "{1}" is set

[Description]

The value that cannot be specified for keyword of the parameter in the definition file is set.

[Parameters]

{0}: command name

{1}: file name

{2}: parameter name

{3}: keyword name

[System Processing]

Processing will be aborted.

[Action]

Take either of the following actions.

- When the keyword is application_name

Correct the definition file or postgresql.auto.conf file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

- Otherwise

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.3.17 MCA00217

{0}: could not use parameter "{3}" because parameter "{2}" is specified in definition file "{1}"

[Description]

This parameter cannot be specified because the parameter with exclusive relationship is specified in the definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: parameter name

{3}: parameter name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.3.18 MCA00218

{0}: value 'fallback' could not be specified for parameter heartbeat_error_action because parameter "{2}" is specified in definition file "{1}"

[Description]

The parameter with exclusive relationship is specified although 'fallback' is specified for heartbeat_error_action.

[Parameters]

{0}: command name

{1}: file name

{2}: parameter name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.3.19 MCA00219

you can switch the connection destination of streaming replication of server "{0}"

[Description]

You can switch the connection destination of streaming replication.

[Parameters]

{0}: server ID

3.3.20 MCA00220

you cannot switch the connection destination of streaming replication of server "{0}"

[Description]

You cannot switch the connection destination of streaming replication.

[Parameters]

{0}: server ID

3.3.21 MCA00221

start to build in standby server "{0}"

[Description]

Start to build in standby server.

[Parameters]

{0}: server ID

3.3.22 MCA00222

failed to build in standby server "{0}"

[Description]

Failed to build in standby server.

[Parameters]

{0}: server ID

[System Processing]

Processing will be aborted.

[Action]

Find the message output before this message from display of own and other server, system log or event log, and then work around according to the Action of the message.

3.3.23 MCA00223

build in standby server "{0}" completed

[Description]

Build in standby server "{0}" completed.

[Parameters]

{0}: server ID

3.3.24 MCA00224

start to switch the connection destination of streaming replication

[Description]

Start to switch the connection destination of streaming replication.

3.3.25 MCA00225

failed to switch the connection destination of streaming replication

[Description]

Failed to switch the connection destination of streaming replication.

[System Processing]

Processing of switching will be aborted.

[Action]

Find the message output before this message from display of own and other server, system log or event log, and then work around according to the Action of the message.

3.3.26 MCA00226

start to switch the connection destination of streaming replication of server "{0}"

[Description]

Start to switch the connection destination of streaming replication.

[Parameters]

{0}: server ID

[Action]

If Mirroring Controller executed switching automatically, find the message output before this message from system log or event log to identify the cause of switching, and then work around according to the Action of the message.

3.3.27 MCA00227

failed to switch the connection destination of streaming replication of server "{0}"

[Description]

Failed to switch the connection destination of streaming replication.

[Parameters]

{0}: server ID

[System Processing]

Processing of switching will be aborted.

[Action]

Find the message output before this message from display own and other server, system log or event log, and then work around according to the Action of the message.

3.3.28 MCA00228

switch the connection destination of streaming replication of server "{0}" completed

[Description]

Switch the connection destination of streaming replication completed.

[Parameters]

{0}: server ID

3.3.29 MCA00229

switch forcibly the connection destination of streaming replication of server "{0}" completed

[Description]

Switch forcibly the connection destination of streaming replication completed.

[Parameters]

{0}: server ID

3.3.30 MCA00230

value of primary_conninfo parameter which is specified in postgresql.auto.conf file of server "{0}" has been updated to the value of parameter "{2}" which is specified in definition file "{1}"

[Description]

Value of primary_conninfo parameter which is specified in postgresql.auto.conf file has been updated.

[Parameters]

{0}: server ID

{1}: file name

{2}: parameter name

3.3.31 MCA00231

failed to update value of primary_conninfo parameter which is specified in postgresql.auto.conf file of server "{0}" detail of cause:"{1}"

[Description]

Failed to update value of primary_conninfo parameter which is specified in postgresql.auto.conf file.

[Parameters]

{0}: server ID

{1}: detail of cause

[Action]

Check the error detail and eliminate causes.

3.3.32 MCA00232

update value of parameter "{2}" to "{3}" which is specified in definition file "{1}" of server "{0}" has completed

[Description]

Update value of parameter which is specified in definition file has completed.

[Parameters]

{0}: server ID

{1}: file name

{2}: parameter name

{3}: primary, standby

3.3.33 MCA00233

failed to update value of parameter "{2}" which is specified in definition file "{1}" of server "{0}"

[Description]

Failed to update value of parameter which is specified in definition file.

[Parameters]

{0}: server ID

{1}: file name

[Action]

Find the message output before this message from display of own and other server, system log or event log, and then work around according to the Action of the message.

3.3.34 MCA00234

failed to check the consistency of LSN between server "{0}" and server "{1}"

[Description]

Failed to check the consistency of LSN between the server.

[Parameters]

{0}: server ID

{1}: server ID

[Action]

Find the message output before this message from display of own and other server, system log or event log, and then work around according to the Action of the message.

3.3.35 MCA00235

failed to update value of parameter "{2}" which is specified in file "{1}" of old candidate primary server "{0}"

[Description]

Failed to update value of parameter which is specified on old candidate primary server.

[Parameters]

{0}: server ID

{1}: file name

{2}: parameter name

[Action]

Either of the following processes will be performed befor restarting OS if automatic start and stop of Mirroring Contoroller has been setting, otherwise building in old candidate primary server to new candidate primary server.

- When update postgresql.auto.conf file has failed

Update connection setting of primary_conninfo parameter which is specified in postgresql.auto.conf file to new candidate primary server by executing ALTER SYSTEM SET statement

- When update "server identifier".conf file has failed

Edit setting of standbycenter_mode parameter in 'standby' which is specified in "server identifier".conf file.

3.3.36 MCA00236

Mirroring Controller of the server "{0}" is not running

[Description]

Mirroring Controller of the server "{0}" is not running.

[Parameters]

{0}: server ID

3.3.37 MCA00237

invalid combination of server own server type:{0} other server type:{1}

[Description]

Invalid combination of server.

[Parameters]

{0}: own server type

{1}: other server type

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description to be set reffering disaster recovery operation of "Cluster Operation Guide (Database Multiplexing)".

3.3.38 MCA00238

{0}: cannot execute detaching forcibly other than on operation center

[Description]

Detaching forcibly needs to be executed only on operation center.

[Parameters]

{0}: command name

[System Processing]

Processing will be aborted.

[Action]

Check whether the server where the command was executed is correct.

3.3.39 MCA00239

standbycenter_mode parameter is specified in the definition file "{0}" of primary server

[Description]

standbycenter_mode parameter is specified in the definition file of primary server.

[Parameters]

{0}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description to be set reffering disaster recovery operation of "Cluster Operation Guide (Database Multiplexing)".

3.3.40 MCA00240

promoted to the primary server, although standbycenter_mode parameter is specified in the definition file "{1}" of the server "{0}"

[Description]

Promoted to the primary server, although standbycenter_mode parameter is specified in the definition file.

[Parameters]

{0}: server ID

{1}: file name

[System Processing]

Stop monitoring.

[Action]

Correct the definition file according to the message and parameter description to be set reffering disaster recovery operation of "Cluster Operation Guide (Database Multiplexing)".

3.3.41 MCA00241

write permission is denied on definition file "{0}"

[Description]

Write permission is denied on definition file.

[Parameters]

{0}: file name

[System Processing]

Processing will be aborted.

[Action]

Re-execute the command, after granting the write permissions to the definition file.

3.3.42 MCA00242

{0}: invalid format value is specified for parameter "{2}" in definition file "{1}"

[Description]

The invalid format value is specified for parameter in definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: parameter name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.3.43 MCA00243

{0}: value is not set to keyword "{1}" of primary_conninfo parameter in postgresql.auto.conf

[Description]

The value is not set to keyword of primary_conninfo parameter in postgresql.auto.conf.

[Parameters]

{0}: command name

{1}: keyword name

[System Processing]

Processing will be aborted.

[Action]

Set the value for the keyword of primary_conninfo parameter in postgresql.auto.conf file according to the description of "Setting Up the Standby Center" for disaster recovery of "Cluster Operation Guide (Database Multiplexing)".

3.3.44 MCA00244

failed to start monitoring of database process

[Description]

Failed to start monitoring of database process.

[System Processing]

Processing will be aborted.

[Action]

Find the message output before this message from display, system log or event log, and then work around according to the Action of the message. After that, execute the mc_ctl command to restart Mirroring Controller.

3.3.45 MCA00245

checking the consistency of LSN between server "{0}" and server "{1}" is not executed because candidate primary server is abnormal

[Description]

Checking the consistency of LSN is not executed because candidate primary server is abnormal.

[Parameters]

{0}: server ID

{1}: server ID

[System Processing]

Processing will be continued.

[Action]

Find the message output before this message from display, system log or event log, and then work around according to the Action of the message, after switching over will be completed. After that, build in old candidate primary server to new candidate primary server.

3.3.46 MCA00252

The parameter node_role is set on Mirroring Controller process, but the parameter sc_port is not set on Mirroring Controller Arbitration process

[Description]

The parameter node_role is set on Mirroring Controller process, but the parameter sc_port is not set on Mirroring Controller Arbitration process.

[System Processing]

Processing will be aborted.

[Action]

Set sc_port parameter in arbitration.conf of Mirroring Controller arbitration process, if operating in Scaleout Controller configuration. If not, delete node_role parameter from "server identifier".conf of Mirroring Controller process.

3.3.47 MCA00253

The parameter node_role is not set on Mirroring Controller process, but the parameter sc_port is set on Mirroring Controller Arbitration process

[Description]

The parameter node_role is not set on Mirroring Controller process, but the parameter sc_port is set on Mirroring Controller Arbitration process.

[System Processing]

Processing will be aborted.

[Action]

Set node_role parameter in a"server identifier".conf of Mirroring Controller process, if operating in Scaleout Controller configuration. If not, delete sc_port parameter from arbitration.conf of Mirroring Controller arbitration process.

3.3.48 MCA00254

The parameter node_role is "{0}" on Mirroring Controller process, but the node role is "{1}" on Mirroring Controller Arbitration process

[Description]

node_role parameter set in Mirroring Controller process is different from node_role parameter set in the arbitration process of the mirroring controller.

[Parameters]

{0}: node role set in Mirroring Controller arbitration process

{1}: node role set in Mirroring Controller process

[System Processing]

Processing will be aborted.

[Action]

Reconfigure node_role parameter set in "server identifier".conf of Mirroring Controller process to match node_role set in arbs.conf of the Scaleout Controller process.

3.3.49 MCA00255

failed to switch over

[Description]

failed to switch over.

3.3.50 MCA00256

only the Mirroring Controller of the coordinator can be restarted

[Description]

only the Mirroring Controller of the coordinator can be restarted.

3.3.51 MCA00257

restarting Mirroring Controller

[Description]

restarting Mirroring Controller.

3.3.52 MCA00258

failed to restart Mirroring Controller target server:"{0}"

[Description]

failed to restart Mirroring Controller.

[Parameters]

{0}: server ID

3.3.53 MCA00259

failed to connect connection manager

[Description]

failed to connect connection manager.

[System Processing]

Processing will be aborted.

[Action]

Check the status of connection manager.

3.3.54 MCA00260

"{0}" has changed.

[Description]

Parameter in "server identifier".conf was changed.

[Parameters]

{0}: changed parameter

[System Processing]

Processing will be aborted.

[Action]

Reset relevant parameter to its pre-modified values and execute the process again.

3.4 Message Numbers Beginning with MCR00000

3.4.1 MCR00001

could not read file "{0}": exception={1}: {2}

[Description]

Could not read the file.

[Parameters]

{0}: file name

{1}: exception type

{2}: exception detail

[System Processing]

Processing will be aborted.

[Action]

Identify the cause according to the message, and then remove it.

3.4.2 MCR00002

{0}: wrong number of server ID in definition file "{1}"

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: command name

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.4.3 MCR00003

{0}: server ID specified in definition file "{1}" too long (max {2} bytes) line {3}

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: max length of server ID

{3}: line number

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.4.4 MCR00004

{0}: wrong server ID in definition file "{1}"

[Description]

Could not use a server ID with same name in network.conf.

[Parameters]

{0}: command name

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter descriptions of "Cluster Operation Guide (Database Multiplexing)".

3.4.5 MCR00005

invalid host name or IP address "{1}" in definition file "{0}"

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: file name

{1}: host name or IP address

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.4.6 MCR00006

{0}: invalid port number in definition file "{1}" line {2}

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: line number

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.4.7 MCR00007

{0}: invalid format specified in definition file "{1}" line {2}

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: line number

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.4.8 MCR00008

{0}: invalid value for parameter "{2}" in definition file "{1}"

[Description]

Invalid parameter was found in definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: parameter name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.4.9 MCR00009

{0}: no value for parameter "{2}" specified in definition file "{1}"

[Description]

Invalid parameter was found in definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: parameter name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.4.10 MCR00010

{0}: unrecognized parameter "{2}" in definition file "{1}"

[Description]

Unrecognized parameter was found in definition file.

[Parameters]

{0}: file name

{1}: command name

{2}: parameter name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.4.11 MCR00011

%s: option "%s" duplicated

[Description]

Certain option is duplicated.

[Parameters]

%s: command

%s: option

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting options.

3.4.12 MCR00012

%s: "%s" option conflicts with "%s" option

[Description]

Options are conflicting.

[Parameters]

%s: command

%s: option

%s: option

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting options.

3.4.13 MCR00013

%s: option requires an argument -- %s

[Description]

No argument specified for the option.

[Parameters]

%s: command

%s: option

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting options.

3.4.14 MCR00014

%s: neither "%s" option nor environment variable "%s" specified

[Description]

Both of required option and equivalent environment variable were not specified.

[Parameters]

%s: command

%s: option

%s: environment variable

[System Processing]

Processing will be aborted.

[Action]

Re-execute after specifying required option or equivalent environment variable.

3.4.15 MCR00015

%s: argument of option "%s" is too long

[Description]

Argument of option is too long.

[Parameters]

%s: command name

%s: option

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting options.

3.4.16 MCR00016

%s: invalid option -- %s

[Description]

Invalid option.

[Parameters]

%s: command name

%s: option

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting options.

3.4.17 MCR00017

%s: option "%s" is required

[Description]

A required option is not specified.

[Parameters]

%s: command name

%s: option

[System Processing]

Processing will be aborted.

[Action]

Specify the required option, and re-execute.

3.4.18 MCR00018

%s: unnecessary operand "%s"

[Description]

Unnecessary operand.

[Parameters]

%s: command name

%s: operand

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting operand.

3.4.19 MCR00019

%s: unrecognized operation mode or no operation mode specified

[Description]

Unrecognized operation mode or no operation mode specified.

[Parameters]

%s: command name

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting or specifying operation mode.

3.4.20 MCR00020

Try "%s --help" for more information.\n

[Description]

--help option can show more additional information.

[Parameters]

%s: command name

[System Processing]

None.

[Action]

Check the message output before this message, and refer to descriptions shown by '--help' option.

3.4.21 MCR00021

%s: out of memory

[Description]

Out of memory error occurred.

[Parameters]

%s: command

[System Processing]

Processing will be aborted.

[Action]

Obtain free memory space by stopping unnecessary processes or changing system settings.

3.4.22 MCR00022

another "{0}" command is running

[Description]

Cannot execute command with this operation mode because another command is running on the same or another server.

[Parameters]

{0}: command name

[System Processing]

Processing will be aborted.

[Action]

There is a case executing another command. Wait for completion of another command on the same or another server, and then re-execute.

In addition, there are the following cases when using mc_arb command.

There is a case under processing of an arbitration by Mirroring Controller Arbitration process. Wait for completion of the processing under operation, and re-execute.

If any of the following cases occurs, there is a possibility that the processing of Mirroring Controller Arbitration process interrupts. Re-execute the mc_arb command after restarting Mirroring Controller Arbitration process.

- When abnormality occurs in the network

- When another server is downed

- When Mirroring Controller Arbitration process is stopped forcibly

3.4.23 MCR00023

Mirroring Controller Arbitration process is already running

[Description]

Mirroring Controller Arbitration process is already running.

[System Processing]

Processing will be aborted.

[Action]

If needed, stop Mirroring Controller Arbitration process, and re-execute.

If could not start although Mirroring Controller Arbitration process not started, refer to the description about workaround for failure of "Cluster Operation Guide (Database Multiplexing)".

3.4.24 MCR00024

cannot execute %s command because Mirroring Controller Arbitration process is not running

[Description]

Cannot execute Mirroring Controller Arbitration process command because Mirroring Controller Arbitration process is not running.

[Parameters]

%s: command name

[System Processing]

Processing will be aborted.

[Action]

Start Mirroring Controller Arbitration process, and re-execute.

3.4.25 MCR00025

timeout waiting for communication with Mirroring Controller Arbitration process

[Description]

Timeout waiting for communication with Mirroring Controller Arbitration process.

[System processing]

Processing will be aborted.

[Action]

Check whether a network error was detected.

If an error was not detected, there is a possibility that the load of the system may be the cause, please re-execute after a while.

3.4.26 MCR00026

could not create PID file of Mirroring Controller Arbitration process detail of cause:"{0}"

[Description]

Could not create PID file of Mirroring Controller Arbitration process.

[Parameters]

{0}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Identify the cause according to the message, and then remove it.

3.4.27 MCR00027

could not remove PID file of Mirroring Controller Arbitration process detail of cause:"%s"

[Description]

Could not remove PID file of Mirroring Controller Arbitration process.

[Parameters]

%s: detail of cause

[Action]

Identify the cause according to the message, and then remove it.

3.4.28 MCR00028

could not read PID file of Mirroring Controller Arbitration process detail of cause:"%s"

[Description]

Could not read PID file of Mirroring Controller Arbitration process.

[Parameters]

%s: detail of cause

[Action]

Identify the cause according to the message, and then remove it.

3.4.29 MCR00029

invalid contents of PID file "%s" of Mirroring Controller Arbitration process

[Description]

The contents of PID file of Mirroring Controller Arbitration process is invalid.

[Parameters]

%s: file name

[System Processing]

Processing will be aborted.

[Action]

The following causes could be considered.

- The file was stored or replaced by mistake

- The file was corrupted

When starting Mirroring Controller Arbitration process, move or remove the file shown in the message.

When stopped Mirroring Controller Arbitration process, find ID of process named "mc_arbiter" and terminate forcibly by using OS command.

3.4.30 MCR00030

unexpected error occurred: {0}

[Description]

An unexpected error occurred.

[Parameters]

{0}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate cause.

If you cannot clear the problem, contact Fujitsu technical support.

3.4.31 MCR00031

system call error occurred:"%s" detail of cause:"%s"

[Description]

System call error occurred.

[Parameters]

%s: system call name

%s: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.4.32 MCR00032

failed to open communication environment detail of cause:"{0}"

[Description]

Failed to open communication environment.

[Parameters]

{0}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

To investigate the cause of the occurrence from the message, and remove cause.

3.4.33 MCR00033

could not read file "{0}": Permission denied

[Description]

No read permissions for the file.

[Parameters]

{0}: file name

[System Processing]

Processing will be aborted.

[Action]

Re-execute the command, after granting the read permissions to the file.

3.4.34 MCR00034

could not read the access privileges of {0}

[Description]

Could not read the access privileges.

[Parameters]

{0}: target directory

[System Processing]

Processing will be aborted.

[Action]

Check the status of the directory and eliminate causes, and then remove it.

3.4.35 MCR00035

failed to set the access privileges of {0}

[Description]

Failed to set the access privileges.

[Parameters]

{0}: target directory

[System Processing]

Processing will be aborted.

[Action]

Check the status of the directory and eliminate causes, and then remove it.

3.4.36 MCR00036

could not get installation path

[Description]

FUJITSU Enterprise Postgres may not be installed.

[System Processing]

Processing will be aborted.

[Action]

Re-install FUJITSU Enterprise Postgres.

3.4.37 MCR00037

could not access "{0}" file detail of cause:"{1}"

[Description]

Could not access the process information file of OS.

[Parameters]

{0}: file name

{1}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.4.38 MCR00038

invalid contents of "{0}" file

[Description]

Invalid contents of /proc/[pid]/status file.

[Parameters]

{0}: file name

[System Processing]

Processing will be aborted.

[Action]

Check mc_arbiter process is existed or not in yourself.

3.4.39 MCR00039

unusable character is included in path "%s" specified as a directory for Mirroring Controller Arbitration process

[Description]

Unusable character is included in path specified as a directory for Mirroring Controller Arbitration process.

[Parameters]

%s: path name

[System Processing]

Processing will be aborted.

[Action]

Correct the path specified as a directory for Mirroring Controller Arbitration process according to the message and mc_arb command descriptions of "Reference".

3.4.40 MCR00040

could not access path "%s" specified as a directory for Mirroring Controller Arbitration process detail of cause:"%s"

[Description]

Could not access path specified as a directory for Mirroring Controller Arbitration process.

[Parameters]

%s: path name

%s: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.4.41 MCR00041

{0}: IP address "{2}" specified for parameter "my_address" in definition file "{1}" is not found

[Description]

IP address specified for my_address parameter in arbitration.conf is not found.

[Parameters]

{0}: command name

{1}: file name

{2}: IP address

[System Processing]

Processing will be aborted.

[Action]

Check whether IP address specified by my_address parameter in arbitration.conf is correct or IP address is valid.

3.4.42 MCR00042

could not access path "{0}" for parameter "{1}" in definition file "{2}" detail of cause:"{3}"

[Description]

Either of the followings has occurred.

- File does not exist

- You do not specify a file

- Could not read the access privileges

[Parameters]

{0}: path name

{1}: parameter name

{2}: file name

{3}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.4.43 MCR00043

{0}: users other than a command executor have the access privileges for definition file "{1}"

[Description]

Users other than a command executor have the access privileges for definition file.

[Parameters]

{0}: command name

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Revoke all the access privileges for any users other than a command executor.

3.4.44 MCR00044

only the owner of definition file "{0}" can execute this command

[Parameters]

{0}: file name

[Description]

Only the owner who created the directory for Mirroring Controller Arbitration process can execute this command.

[System Processing]

Processing will be aborted.

[Action]

Re-execute the command by the owner who created the directory for Mirroring Controller Arbitration process.

3.4.45 MCR00045

starting Mirroring Controller Arbitration process

[Description]

Starting Mirroring Controller Arbitration process.

3.4.46 MCR00046

Mirroring Controller Arbitration process started

[Description]

Mirroring Controller Arbitration process started.

3.4.47 MCR00047

failed to start Mirroring Controller Arbitration process

[Description]

Failed to start Mirroring Controller Arbitration process.

[System Processing]

Processing will be aborted.

[Action]

Work around according to the Action of the message output before this message.

On Windows, if there is no message outputted before this message, please refer to the message outputted to an event log.

MCR00048

failed to report Mirroring Controller Arbitration process status

[Description]

Failed to report Mirroring Controller Arbitration process status.

[System Processing]

Processing will be aborted.

[Action]

Work around according to the Action of the message output before this message.

On Windows, if there is no message outputted before this message, please refer to the message outputted to an event log.

3.4.48 MCR00049

stopping Mirroring Controller Arbitration process

[Description]

Stopping Mirroring Controller Arbitration process.

3.4.49 MCR00050

Mirroring Controller Arbitration process stopped

[Description]

Mirroring Controller Arbitration process stopped.

3.4.50 MCR00051

could not stop Mirroring Controller Arbitration process because database server connects

[Description]

Could not stop Mirroring Controller Arbitration process because database server connects.

[System Processing]

Processing will be aborted.

[Action]

Execute mc_arb status command to find connected database server.

Re-execute after stopping Mirroring Controller on database server.

3.4.51 MCR00052

failed to stop Mirroring Controller Arbitration process

[Description]

Failed to stop Mirroring Controller Arbitration process.

[System Processing]

Processing will be aborted.

[Action]

Identify the cause from system log or event log on the target server, and work around.

3.4.52 MCR00053

stopping Mirroring Controller Arbitration process forcibly

[Description]

Stopping Mirroring Controller Arbitration process forcibly.

3.4.53 MCR00054

Mirroring Controller Arbitration process stopped forcibly

[Description]

Mirroring Controller Arbitration process stopped forcibly.

3.4.54 MCR00055

failed to stop Mirroring Controller Arbitration process forcibly

[Description]

Failed to stop Mirroring Controller Arbitration process forcibly.

[System Processing]

Processing will be aborted.

[Action]

Check [Action] of the message output before this message,and re-execute.

If re-execution fails, terminate forcibly mc_arbiter process with OS command.

3.4.55 MCR00056

database server "{0}" requested to arbitrate for database server "{1}" status

[Description]

Accepts a request to arbitrate from the database server which has detected an error.

[Parameters]

{0}: server ID

{1}: server ID

3.4.56 MCR00057

arbitrating the status of database server "{0}"

[Description]

Arbitrating for the status of the database server which was detected an error.

[Parameters]

{0}: server ID

[System Processing]

Arbitrating for the status of the database server which was detected an error.

3.4.57 MCR00058

received response from database server "{0}"

[Description]

A server is running normally.

[Parameters]

{0}: server ID

[System Processing]

Return result to database server.

3.4.58 MCR00059

could not receive response from database server "{0}"

[Description]

Detected the database server to be abnormal.

[Parameters]

{0}: server ID

[System Processing]

Execute fencing command.

3.4.59 MCR00060

rejected a request to arbitrate from database server "{0}"

[Description]

Either of the followings has occurred.

- Fencing command was executed just before

- The arbitration server was in the stop processing

[Parameters]

{0}: server ID

[System Processing]

Rejected a request to arbitrate from the database server.

3.4.60 MCR00061

database server "{0}" requested to fence database server "{1}"

[Description]

Accepted a request to fence from the database server which has detected an error.

[Parameters]

{0}: server ID

{1}: server ID

3.4.61 MCR00062

executing fencing command

[Description]

Executing fencing command.

[System processing]

Executing fencing command.

3.4.62 MCR00063

fencing command for database server "{0}" succeeded: result:"{1}"

[Description]

Fencing command for database server succeeded.

[parameters]

{0}: server ID

{1}: return code of command

[System processing]

Returning result to database server.

3.4.63 MCR00064

fencing command for database server "{0}" failed: result:"{1}"

[Description]

Fencing command for database server failed.

[parameters]

{0}: server ID

{1}: return code of command

[System processing]

Returning result to database server.

3.4.64 MCR00065

timeout waiting for the fencing command

[Description]

Timeout waiting for the fencing command.

[System processing]

Processing will be aborted.

[Action]

The value of fencing_command_timeout parameter in arbitration.conf is too short.

Review and extend the value of fencing_command_timeout parameter in arbitration.conf.

When find process ID of fencing command, terminate forcibly by using OS command.

3.4.65 MCR00066

rejected a request to fence from database server "{0}"

[Description]

Either of the followings is esteemed.

- The arbitration server has been executing arbitration process

- Fencing command has been executed just before

- The arbitration server was in the stop processing

[Parameters]

{0}: server ID

[System Processing]

Rejected a request to fence from database server.

3.4.66 MCR00067

rejected a request to fence from database server "{0}" because the database server is a fencing target

[Description]

The following requirements shouldn't be executed if it is requested from the database server which is a fencing target.

- request to arbitrate

- request to fence

- request to disable automatically switch

[Parameters]

{0}: server ID

[Action]

Processing will be aborted.

3.4.67 MCR00068

database server "{0}" requested to disable standby server "{1}" to automatically switch

[Description]

Accepted a request to disable the standby server to automatically switch.

[Parameters]

{0}: sever ID

{1}: sever ID

3.4.68 MCR00069

requested standby server "{0}" to disable automatically switching

[Description]

Requested the standby server to disable automatically switching.

[Parameters]

{0}: server ID

3.4.69 MCR00070

disabled standby server "{0}" to automatically switch

[Description]

Disabled the standby server to automatically switch.

[Parameters]

{0}: server ID

3.4.70 MCR00071

failed to disable standby server "{0}" to automatically switch

[Description]

Failed to disable the standby server to automatically switch.

[Parameters]

{0}: server ID

[System processing]

Executing fencing command.

3.4.71 MCR00072

omitted to disable standby server "{0}" to automatically switch

[Description]

It is not necessary to disable the standby server to automatically switch because a fencing for this database server is executing.

[Parameters]

{0}: server ID

[System processing]

Processing will be aborted.

3.4.72 MCR00073

connection was requested from database server "{0}"

[Description]

Connection was requested from database server.

[Parameters]

{0}: server ID

[Description]

Try connection to database server.

3.4.73 MCR00074

succeeded in connection with Mirroring Controller process of database server "{0}"

[Description]

Succeeded in connection with Mirroring Controller process of database server.

[Parameters]

{0}: server ID

3.4.74 MCR00075

rejected a request to connect from database server "{0}" because of during the fencing command execution

[Description]

Rejected a request to connect from database server because of during the fencing command execution.

[Parameters]

{0}: server ID

[Action]

Rejected a request to connect from database server.

3.4.75 MCR00076

disconnected from the database server "{0}"

[Description]

Disconnected from the database server.

[Parameters]

{0}: server ID

3.4.76 MCR00077

the disallowed IP address "{0}" tried to access

[Description]

The IP address which is different from the IP address in network.conf tried to access.

[parameters]

{0}: IP Adresses

[System Processing]

Connection will be rejected.

3.4.77 MCR00078

tried to access from database srever with installed different product version IP address:"{0}"

[Description]

tried to access from database server with installed different product version.

[parameters]

{0}: IP address

[System Processing]

Connection will be rejected

[Action]

Match the product version level between arbitration server and database server.

3.4.78 MCR00079

the invalid server ID tried to access IP address:"{0}"

[Description]

The server ID which is different from the server ID in network.conf tried to access.

[Parameters]

{0}: IP address

[System Processing]

Connection will be rejected.

3.4.79 MCR00080

the combination of server IDs of arbitration server is incompatible with that of database server "{0}"

[Description]

The combination of server IDs of arbitration server is incompatible with that of the database server.

[Parameters]

{0}: server ID

[System Processing]

Processing will be aborted.

[Action]

Check the network.conf of both database server and arbitration sever, and correct the file.

3.4.80 MCR00081

the invalid packet received

[Description]

The invalid packet received.

[System Processing]

Connection will be rejected.

3.4.81 MCR00082

{0}: invalid server kind specified in definition file "{1}" line {2}

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: command name

{1}: file name

{2}: line number

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "Cluster Operation Guide (Database Multiplexing)".

3.4.82 MCR00083

%s: invalid argument for option %s

[Description]

Invalid argument for option.

[Parameters]

%s: command name

%s: option

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting options.

3.4.83 MCR00084

installation environment is destroyed

[Description]

FUJITSU Enterprise Postgres may not be installed correctly or may be destroyed.

[System Processing]

Processing will be aborted.

[Action]

Re-install FUJITSU Enterprise Postgres.

3.4.84 MCR00085

no authority to execute this command

[Description]

Only the user who possesses an administrative authority can run this command.

[System Processing]

Processing will be aborted.

[Action]

Invoke the administrator's prompt, and re-execute this command.

3.4.85 MCR00086

failed to start service "{0}" detail of cause:"{1}"

[Description]

Failed to start service.

[Parameters]

{0}: Service name

{1}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.4.86 MCR00087

service "{0}" is not registered

[Description]

Service is not registered.

[Parameters]

{0}: Service name

[System Processing]

Processing will be aborted.

[Action]

Register service, and re-execute.

3.4.87 MCR00088

service "%s" for Mirroring Controller Arbitration process was registered

[Description]

Service for Mirroring Controller Arbitration process was registered with Windows service.

[Parameters]

%s: Service name

3.4.88 MCR00089

failed to register service "%s" detail of cause:"%s"

[Description]

An error occurred during registration of service.

[Parameters]

%s: Service name

%s: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.4.89 MCR00090

service name "%s" is already in use

[Description]

Service name is already in use.

[Parameters]

%s: Service name

[System Processing]

Processing will be aborted.

[Action]

Check the service name, and re-execute.

3.4.90 MCR00091

service "%s" for Mirroring Controller Arbitration process was unregistered

[Description]

Service for Mirroring Controller Arbitration process was unregistered from Windows service.

[Parameters]

%s: Service name

3.4.91 MCR00092

failed to unregister service "%s" detail of cause:"%s"

[Description]

An error occurred during deregistration of service.

[Parameters]

%s: Service name

%s: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

3.4.92 MCR00093

an error occurred in Mirroring Controller Arbitration process: {0}

[Description]

An error occurred in Mirroring Controller Arbitration process.

[Parameters]

{0}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

If you cannot clear the problem, contact Fujitsu technical support.

After removing the cause of errors, execute mc_arb command with the -e option to stop Mirroring Controller Arbitration process forcibly.

After that, execute the mc_arb command to start Mirroring Controller Arbitration process.

3.4.93 MCR00094

value of heartbeat_interval in arbitration definition file "{1}" is too large compared with value of parameters for abnormality monitoring of operating system or server in server definition file of database server "{0}"

[Description]

Because the value of heartbeat_interval parameter in the arbitration definition file is too large compared with the value of parameters for abnormality monitoring of the operating system or server in the server definition file of the database server, the arbitration for the target database server might be delayed.

[Parameters]

{0}: server ID

{1}: file name

[System Processing]

Continues processing.

[Action]

Correct the parameters for abnormality monitoring of the operating system or server according to the message and "Tuning for Optimization of Degradation Using Abnormality Monitoring With the Arbitration Server" of "Cluster Operation Guide (Database Multiplexing)".

After that, execute the mc_arb command to restart Mirroring Controller Arbitration process.

3.4.94 MCR00095

start to monitor of database server "{0}"

[Description]

Start to monitor of the database server.

[Parameters]

{0}: server ID

3.4.95 MCR00096

failed to start monitoring of database server "{0}"

[Description]

Failed to start monitoring of the database server.

[Parameters]

{0}: server ID

[System Processing]

Processing will be aborted.

[Action]

Find the message output before this message from display, system log or event log, and then work around according to the Action of the message.

3.4.96 MCR00097

stop Mirroring Controller Arbitration process forcibly because an error has occurred in Mirroring Controller Arbitration process: {0}

[Description]

Stop Mirroring Controller Arbitration process forcibly because an error has occurred in Mirroring Controller Arbitration process

[Parameters]

{0}: detail of cause

[System Processing]

Mirroring Controller Arbitration process will be stopped forcibly.

[Action]

Check the error detail and eliminate causes.

If you cannot clear the problem, contact Fujitsu technical support.

After removing the cause of errors, execute the mc_arb command to start Mirroring Controller Arbitration process.

3.4.97 MCR00098

stop monitoring of database server "{0}"

[Description]

Stop monitoring of the database server.

[Parameters]

{0}: server ID

3.4.98 MCR00099

detected an error on database server "{0}"

[Description]

Either of the followings has occurred.

- The database server is downed

- The arbitration network is abnormal

[Parameters]

{0}: server ID

[System Processing]

Continues processing.

3.5 Message Numbers Beginning with MCR00100

3.5.1 MCR00100

detected recovery of database server "{0}"

[Description]

detected recovery of the database server.

[Parameters]

{0}: server ID

3.5.2 MCR00101

arbitration for database server "{0}" has been delayed

[Description]

Either of the followings has occurred.

- The parameters for abnormality monitoring of the operating system or server are not optimally tuned

- The arbitration server is not responding

[Parameters]

{0}: server ID

[System Processing]

Continues processing.

[Action]

Take either of the following actions.

- When parameters for abnormality monitoring of the operating system or server are not optimally tuned

Correct the value of parameters for abnormality monitoring of the operating system or server according to the message and "Tuning for Optimization of Degradation Using Abnormality Monitoring With the Arbitration Server" of "Cluster Operation Guide (Database Multiplexing)".

After that, execute the mc_ctl command to stop Mirroring Controller, and execute the mc_arb command to stop the Mirroring Controller Arbitration process.

After that, execute the mc_arb command to start the Mirroring Controller Arbitration process, and execute the mc_ctl command to start Mirroring Controller.

- When the arbitration server is not responding

Identify the cause, and then remove it.

3.5.3 MCR00102

connection was requested from Scaleout Controller

[Description]

connection was requested from Scaleout Controller.

[System Processing]

Try connection to Scaleout Controller.

3.5.4 MCR00103

succeeded in connection with Scaleout Controller process

[Description]

succeeded in connection with Scaleout Controller process.

3.5.5 MCR00104

node role of Mirroring Controller Arbitration process has been set to "{0}"

[Description]

node role of Mirroring Controller Arbitration process has been set.

[Parameters]

{0}: node role

3.5.6 MCR00105

The parameter sc_port is not set on Mirroring Controller Arbitration process, but the parameter node_role is set on Mirroring Controller process

[Description]

The parameter sc_port is not set on Mirroring Controller Arbitration process, but the parameter node_role is set on Mirroring Controller process.

[System Processing]

Processing will be aborted.

[Action]

Set sc_port parameter in arbitration.conf of Mirroring Controller arbitration process, if operating in Scaleout Controller configuration. If not, delete node_role parameter from "server identifier".conf of Mirroring Controller process.

3.5.7 MCR00106

The node role is "{0}" on Mirroring Controller Arbitration process, but the parameter node_role is "{1}" on Mirroring Controller process

[Description]

node_role parameter set in Mirroring Controller process is different from node_role parameter set in the arbitration process of the mirroring controller.

[Parameters]

{0}: node role set in Mirroring Controller arbitration process

{1}: node role set in Mirroring Controller process

[System Processing]

Processing will be aborted.

[Action]

Reconfigure node_role parameter set in "server identifier".conf of Mirroring Controller process to match node_role set in arbs.conf of the Scaleout Controller process.

3.5.8 MCR00107

The parameter sc_port is set on Mirroring Controller Arbitration process, but the parameter node_role is not set on Mirroring Controller process

[Description]

The parameter sc_port is set on Mirroring Controller Arbitration process, but the parameter node_role is not set on Mirroring Controller process.

[System Processing]

Processing will be aborted.

[Action]

Set node_role parameter in a"server identifier".conf of Mirroring Controller process, if operating in Scaleout Controller configuration. If not, delete sc_port parameter from arbitration.conf of Mirroring Controller arbitration process.

3.5.9 MCR00108

timeout waiting for communication with Scaleout Controller process

[Description]

timeout waiting for communication with Scaleout Controller process.

[System Processing]

Processing will be aborted.

[Action]

Check whether Scaleout Controller process is in a normal state.

3.5.10 MCR00109

cannot send node status because Scaleout Controller process is not running

[Description]

cannot send node status because Scaleout Controller process is not running.

[Action]

Check the status of Scaleout Controller process.

3.5.11 MCR00110

The node role is "{0}" on Mirroring Controller Arbitration process, but the node role is "{1}" on Scaleout Controller process

[Description]

node_role parameter set in the arbitration process of the mirroring controller is different from node_role parameter set in Scaleout Controller process.

[Parameters]

{0}: node role set in Mirroring Controller arbitration process

{1}: node role set in Mirroring Controller process

[System Processing]

Processing will be aborted.

[Action]

Reconfigure node_role parameter set in "server identifier".conf of Mirroring Controller process to match node_role set in arbs.conf of the Scaleout Controller process.

3.5.12 MCR00111

disconnected from the Scaleout Controller

[Description]

disconnected from the Scaleout Controller.

3.5.13 MCR00112

{0}: wrong number of coordinator node in definition file "{1}"

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: command name

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "manual".

3.5.14 MCR00113

{0}: no data node in definition file "{1}"

[Description]

Invalid descriptions were found in definition file.

[Parameters]

{0}: command name

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the definition file according to the message and parameter description of "manual".

3.5.15 MCR00114

Scaleout Controller process is already running

[Description]

Scaleout Controller process is already running.

[System Processing]

Processing will be aborted.

[Action]

If needed, stop Scaleout Controller process, and re-execute.

If could not start although Scaleout Controller process not started, refer to the description about workaround for failure of "manual".

3.5.16 MCR00115

cannot execute "{0}" command because Scaleout Controller process is not running

[Description]

Cannot execute Scaleout Controller process command because Scaleout Controller process is not running.

[Parameters]

{0}: command name

[System Processing]

Processing will be aborted.

[Action]

Start Scaleout Controller process, and re-execute.

3.5.17 MCR00116

timeout waiting for communication with Scaleout Controller process

[Description]

timeout waiting for communication with Scaleout Controller process.

[System Processing]

Processing will be aborted.

[Action]

Check whether a network error was detected.

If an error was not detected, there is a possibility that the load of the system may be the cause, please re-execute after a while.

3.5.18 MCR00117

could not create PID file of Scaleout Controller process detail of cause:"{0}"

[Description]

Could not create PID file of Scaleout Controller process.

[Parameters]

{0}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Identify the cause according to the message, and then remove it.

3.5.19 MCR00118

could not remove PID file of Scaleout Controller process detail of cause:"%s"

[Description]

Could not remove PID file of Scaleout Controller process.

[Parameters]

%s: detail of cause

[Action]

Identify the cause according to the message, and then remove it.

3.5.20 MCR00119

could not read PID file of Scaleout Controller process detail of cause:"{0}"

[Description]

could not read PID file of Scaleout Controller process.

[Parameters]

%s: detail of cause

[Action]

Identify the cause according to the message, and then remove it.

3.5.21 MCR00120

invalid contents of PID file "{0}" of Scaleout Controller process

[Description]

The contents of PID file of Scaleout Controller process is invalid.

[Parameters]

{0}: file name

[System Processing]

Processing will be aborted.

[Action]

The following causes could be considered.

- The file was stored or replaced by mistake

- The file was corrupted

When starting Scaleout Controller process, move or remove the file shown in the message.

When stopped Scaleout Controller process, find ID of process named "sc_arbiter" and terminate forcibly by using OS command.

3.5.22 MCR00121

unusable character is included in path "%s" specified as a directory for Scaleout Controller process

[Description]

Unusable character is included in path specified as a directory for Scaleout Controller.

[Parameters]

%s: path name

[System Processing]

Processing will be aborted.

[Action]

Correct the path specified as a directory for Scaleout Controller according to the message and sc_ctl command descriptions of "Reference".

3.5.23 MCR00122

could not access path "%s" specified as a directory for Scaleout Controller process detail of cause:"%s"

[Description]

Could not access path specified as a directory for Scaleout Controller process.

[Parameters]

%s: path name

%s: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Identify the cause according to the message, and then remove it.

3.5.24 MCR00123

The node role is "{0}" on Scaleout Controller process, but the node role is "{1}" on Mirroring Controller Arbitration process

[Description]

The node role of each node set in the Scaleout Controller process differs from the node type set in the each Mirroring Controller arbitration process.

[Parameters]

{0}: node role set in arbs.conf

{1}: node_role set in "server identifier".conf

[System Processing]

Processing will be aborted.

[Action]

Re-execute after correcting the setting value.

3.5.25 MCR00124

an error occurred in Scaleout Controller process: {0}

[Description]

an error occurred in Scaleout Controller process.

[Parameters]

{0}: detail of cause

[System Processing]

Processing will be aborted.

[Action]

Check the error detail and eliminate causes.

If you cannot clear the problem, contact Fujitsu technical support.

After removing the cause of errors, execute sc_ctl command to stop Scaleout Controller process.

If re-execution fails, terminate forcibly sc_arbiter process with OS command.

After that, execute the sc_ctl command to start Scaleout Controller process.

3.5.26 MCR00125

starting Scaleout Controller process

[Description]

starting Scaleout Controller process.

3.5.27 MCR00126

starting Mirroring Controller Arbitration Process: "{0}"

[Description]

starting Mirroring Controller Arbitration Process.

[Parameters]

{0}: directory of relevant Mirroring Controller arbitration process

3.5.28 MCR00127

requesting arbitration server "{0}" to connect

[Description]

requesting arbitration server to connect.

[Parameters]

{0}: directory of relevant Mirroring Controller arbitration process

[System Processing]

Requesting arbitration server to connect.

3.5.29 MCR00128

trying to connect to arbitration server "{0}"

[Description]

trying to connect to arbitration server.

[Parameters]

{0}: directory of relevant Mirroring Controller arbitration process

[System Processing]

Trying to connect to arbitration server until success.

3.5.30 MCR00129

succeeded in connection with arbitration server "{0}"

[Description]

succeeded in connection with arbitration server.

[Parameters]

{0}: directory of relevant Mirroring Controller arbitration process

3.5.31 MCR00130

failed to connect to arbitration server "{0}" event:"{1}"

[Description]

Either of the followings has occurred.

- Mirroring Controller Arbitration process is not running or in the stop processing

- Mirroring Controller Arbitration process or arbitration server detects an error

[Parameters]

{0}: server ID

{1}: "timeout" or "communication error"

[System Processing]

Processing will be aborted.

[Action]

Check the following and identify the cause, and eliminate cause.

- Mirroring Controller Arbitration process staring status

- the message in arbitration server

3.5.32 MCR00131

timeout waiting for communication with Mirroring Controller Arbitration process server:"{0}"

[Description]

Timeout waiting for communication between Scaleout Controller process and Mirroring Controller Arbitration process.

[Parameters]

{0}: directory of relevant Mirroring Controller arbitration process

[System Processing]

Try to connect to arbitration server.

[Action]

Check whether Mirroring Controller arbitration process is in a normal state.

3.5.33 MCR00132

communication error with the arbitration server "{0}" occurred

[Description]

Communication error with the arbitration server occurred.

[Parameters]

{0}: directory of relevant Mirroring Controller arbitration process

[System Processing]

Trying to connect.

[Action]

Check whether Mirroring Controller Arbitration process is running.

Identify the cause from messages on system log or event log in arbitration server, or Scaleout Controller server, and work around.

3.5.34 MCR00133

failed to start Scaleout Controller because of could not connect to arbitration server "{0}"

[Description]

failed to start Scaleout Controller because of could not connect to arbitration server.

[Parameters]

{0}: directory of relevant Mirroring Controller arbitration process

[System Processing]

Processing will be aborted.

[Action]

Work around according to the Action of the message output before this message.

3.5.35 MCR00134

values of parameters for abnormality monitoring of operating system or server in server definition file "{1}" are too small for value of heartbeat_interval in arbitration definition file of arbitration server "{0}"

[Description]

Because the values of parameters for abnormality monitoring of the operating system or server in the server definition file are too small compared with the value of heartbeat_interval in the arbitration definition file of the arbitration server, the arbtration for the target database server might be delayed.

[System Processing]

{0}: server ID

{1}: file name

[System Processing]

Processing will be aborted.

[Action]

Correct the value of parameters for abnormality monitoring of the operating system or server according to the message and "Tuning for Optimization of Degradation Using Abnormality Monitoring With the Arbitration Server" of "Cluster Operation Guide (Database Multiplexing)".

After that, execute the sc_ctl command to restart Scaleout Controller.

3.5.36 MCR00135

Scaleout Controller process started

[Description]

Scaleout Controller process started.

3.5.37 MCR00136

failed to start Scaleout Controller process

[Description]

failed to start Scaleout Controller process.

[System Processing]

Processing will be aborted.

[Action]

Work around according to the Action of the message output before this message.

3.5.38 MCR00137

could not execute "mc_arb status -M {0}"

[Description]

could not execute status mode of mc_arb command.

[Parameters]

{0}: directory of relevant Mirroring Controller arbitration process

[System Processing]

Processing will be aborted.

[Action]

Work around according to the Action of the message output before this message.

3.5.39 MCR00138

failed to report Scaleout Controller process status

[Description]

failed to report Scaleout Controller process status.

[System Processing]

Processing will be aborted.

[Action]

Work around according to the Action of the message output before this message.

3.5.40 MCR00139

stopping Scaleout Controller process

[Description]

stopping Scaleout Controller process.

3.5.41 MCR00140

stopping Mirroring Controller Arbitration Process: {0}

[Description]

stopping Mirroring Controller Arbitration Process.

3.5.42 MCR00141

could not stop Mirroring Controller Arbitration process "{0}"

[Description]

could not stop Mirroring Controller Arbitration process.

[Parameters]

{0}: directory of relevant Mirroring Controller arbitration process

[System Processing]

Processing will be aborted.

[Action]

Work around according to the Action of the message output before this message.

3.5.43 MCR00142

Scaleout Controller process stopped

[Description]

Scaleout Controller process stopped.

3.5.44 MCR00143

could not stop Scaleout Controller process because arbitration server connects

[Description]

could not stop Scaleout Controller process because arbitration server connects.

[System Processing]

Processing will be aborted.

[Action]

Arbitration server communicating with the Scaleout Controller process is found, please re-execute after a while.

3.5.45 MCR00144

failed to stop Scaleout Controller process

[Description]

failed to stop Scaleout Controller process.

[System Processing]

Processing will be aborted.

[Action]

Identify the cause from system log on the target server, and work around.

3.5.46 MCR00145

cannot execute "{0}" command because Mirroring Controller Arbitration process is not running

[Description]

Cannot execute Mirroring Controller Arbitration process command because Mirroring Controller Arbitration process is not running.

[Parameters]

{0}: command name

[Action]

Start Mirroring Controller Arbitration process, and re-execute.

3.5.47 MCR00146

restarting Scaleout Controller process

[Description]

restarting Scaleout Controller process.

3.5.48 MCR00147

"{0}" has changed

[Description]

Parameter in sc.conf was changed.

[Parameters]

{0}: changed parameter

[System Processing]

Processing will be aborted.

[Action]

Reset relevant parameter to its pre-modified values and execute the process again.

3.5.49 MCR00148

failed to restart Scaleout Controller

[Description]

failed to restart Scaleout Controller.

[System Processing]

Processing will be aborted.

3.5.50 MCR00149

add node "{0}" as a managed node by Scaleout Controller Process

[Description]

add node as a managed node by Scaleout Controller Process

[Parameters]

{0}: node name

3.5.51 MCR00150

both datanode "{0}" and coordinator node will perform failover

[Description]

both datanode and coordinator node will perform failover.

[Parameters]

{0}: node name

3.5.52 MCR00151

cannot continue operation on this center

[Description]

cannot continue operation on this center

[Action]

Switch center.

3.5.53 MCR00152

cannot find primary server on coordinator node

[Description]

cannot find primary server on coordinator node.

3.5.54 MCR00153

cannot solve network disconnection between node:"{0}" and node:"{1}"

[Description]

The arbitration process by Scaleout Controller process could not solve the inter-node network disconnection condition.

[Parameters]

{0}: nodename

{1}: nodename

3.5.55 MCR00154

cannot solve network trouble between nodes, need to switch center

[Description]

cannot solve network trouble between nodes, need to switch center.

[Action]

Switch center.

3.5.56 MCR00155

coordinator node will perform failover

[Description]

coordinator node will perform failover.

3.5.57 MCR00156

datanode "{0}" is being detached

[Description]

the target datanode is being detached

[Parameters]

{0}: nodename

3.5.58 MCR00157

datanode "{0}" will perform failover

[Description]

the target datanode will perform failover

[Parameters]

{0}: nodename

3.5.59 MCR00158

failed to perform failover on "{0}"

[Description]

failed to perform failover on the target node.

[Parameters]

{0}: nodename

3.5.60 MCR00159

failover status was not true even the node did failover

[Description]

failover status was not true even the node did failover.

3.5.61 MCR00160

got connection status of unrecognaized coordinator server

[Description]

got connection status of unrecognaized coordinator server.

3.5.62 MCR00161

Invalid server type

[Description]

Invalid server type.

3.5.63 MCR00162

Invalid server

[Description]

Invalid server.

3.5.64 MCR00163

no server with port "{0}" found

[Description]

no server with the specified port found.

[Parameters]

{0}: port number

3.5.65 MCR00164

succeeded in performing failover node "{0}"

[Description]

succeeded in performing failover node.

[Parameters]

{0}: nodename

3.5.66 MCR00165

unknown coordinator server connected

[Description]

unknown coordinator server connected.

3.5.67 MCR00166

could not execute because Mirroring Controller Arbitration process of the node "{0}" is not running

[Description]

could not execute because Mirroring Controller Arbitration process of the target node is not running.

[Parameters]

{0}: directory of relevant Mirroring Controller arbitration process

[System Processing]

Processing will be aborted.

[Action]

Re-execute after starting the specified Mirroring Controller arbitration process.

3.5.68 MCR00167

connection was requested from arbiter server "{0}"

[Description]

connection was requested from arbiter server.

[Parameters]

{0}: directory of relevant Mirroring Controller arbitration process

3.5.69 MCR00168

disconnected from arbitration server "{0}"

[Description]

disconnected from arbitration server.

[Parameters]

{0}: directory of relevant Mirroring Controller arbitration process

Index

[E]

Error type....2

[F]

Format of messages output to the server message log....1
Format of messages returned to an application....1

[M]

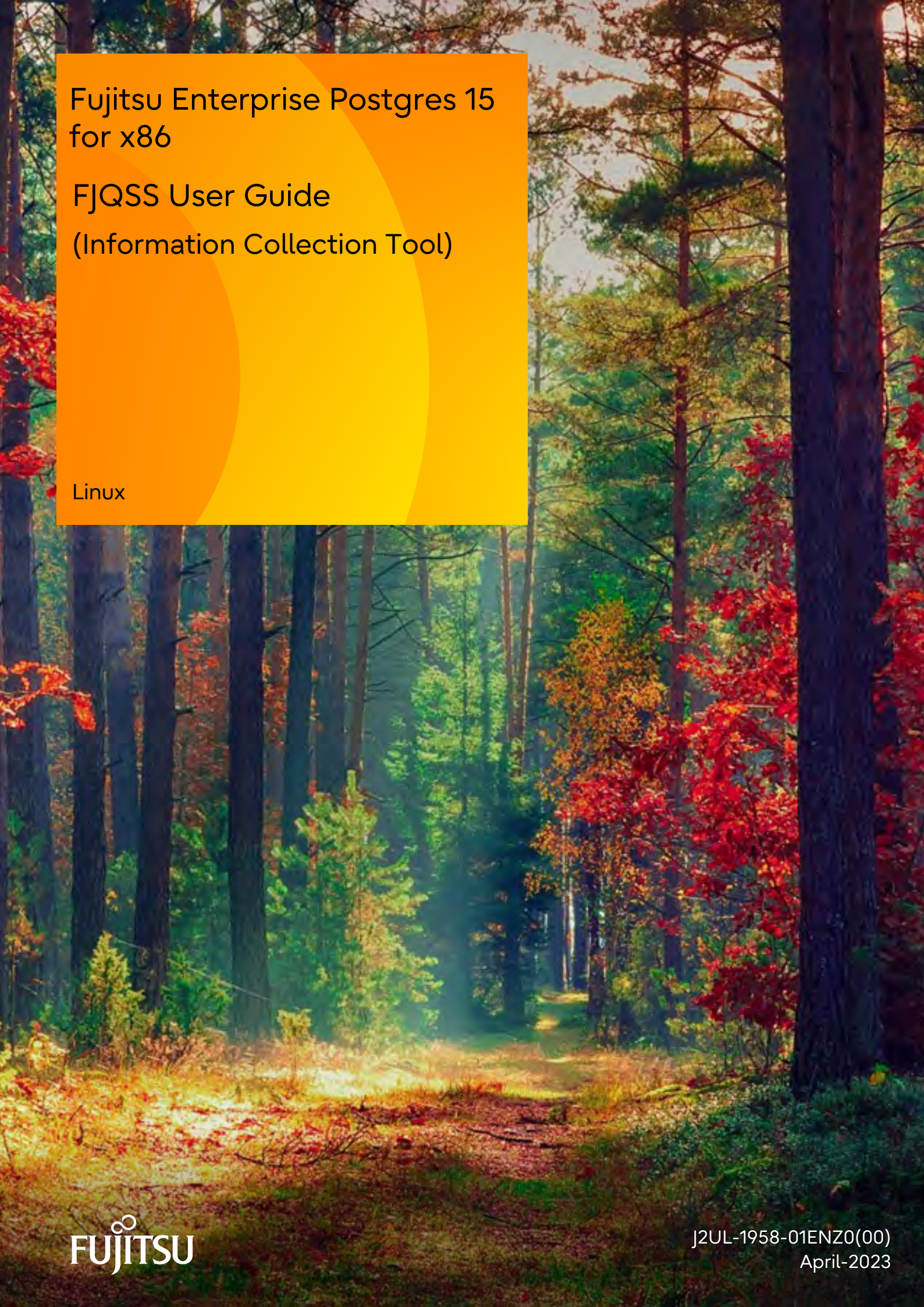
Message Format....1
Message text....2
Mirroring Controller Message Format....2

[N]

Notes on monitoring messages output to the server message log1
Notes on monitoring messages returned to an application....1

[O]

Overview of Messages....1



Fujitsu Enterprise Postgres 15 for x86

FJQSS User Guide

(Information Collection Tool)

Linux

FUJITSU

J2UL-1958-01ENZ0(00)
April-2023

Preface

This manual describes information to use the outline and the tool of FJQSS (Information Collection Tool).

Purpose of this Manual

The purpose of this manual is to be able to gather the collected information by using FJQSS (Information Collection Tool) when trouble occurs.

Reader of this Manual

This manual is targeted at the user who gathers the collected information when the trouble occurs.

You can obtain information on the method of gathering the function and the collected information of FJQSS (Information Collection Tool) obtain by referring to this manual.

Structure of this Manual

The structure of this manual is as follows:

Chapter 1 outline

Chapter 2 FJQSS (Information Collection Tool)

Appendix A Messages and error codes

Notations

The following notations are used in this manual;

Symbols used in this manual

The following symbols are used in this manual.

Example:[Enter]key

Symbols used in command

This subsection describes the symbols used in the examples of commands.

Example:

```
fjqss_collect [-s]
```

Meaning of Symbols

Symbol	Meaning
[]	The item enclosed with this sign is shown and omissible is shown.

Abbreviations

The following abbreviations are used in this manual:

Operating Systems

Full Name	Abbreviations
Oracle Solaris 9	Solaris 9
Oracle Solaris 10	Solaris 10
Oracle Solaris 11	Solaris 11
Oracle Solaris	Solaris
Red Hat Enterprise Linux 6 (for x86)	RHEL6(x86)
Red Hat Enterprise Linux 6 (for Intel64)	RHEL6(Intel64)
Red Hat Enterprise Linux 5 (for x86)	RHEL5(x86)
Red Hat Enterprise Linux 5 (for Intel64)	RHEL5(Intel64)

Export Management Regulations

Our documentation may include special technology based on Foreign Exchange and Foreign Trade Control Law.

In case you export the relevant documentation or to provide any overseas resident with the relevant documentation, permission based on the above law is necessary.

Trademarks and Screenshots

Oracle and Java are registered trademarks of Oracle Corporation, its subsidiaries and its affiliated companies in the United States and other countries.

The names of companies and products, etc. in the sentences inside can be trademarks or registered trademarks of each company.

Date of Publication

Date of Publication and Version	Manual Code
November 2012 Version1	J2S2-1625-01ENZ2(00)
April 2013 Version1.1.	J2S2-1625-02ENZ2(00)
October 2013 Version 1.2	J2S2-1625-03ENZ2(00)
July 2014 Version 1.3	J2UL-1958-01EN2Z(00)

Copyright Notice

Copyright 2014 FUJITSU LIMITED

Contents

Chapter 1 Outline 1
1.1 FJQSS (Information Collection Tool) 1

Chapter 2 Collecting the information 2
2.1 Collection method 2
2.1.1 Collecting with showing menu 2
2.1.2 Collecting without showing menu 4
2.1.3 Getting the product identification name 5
2.2 Checking the collected information 6
2.3 Changing a path of the output directory 6

Appendix A Messages and error codes 9
A.1 Error Messages 9
A.2 Error Codes 10

Chapter 1 Outline

This chapter explains the outline of FJQSS (Information Collection Tool).

1.1 FJQSS (Information Collection Tool)

This appendix explains FJQSS (Information Collection Tool) that archives the information for investigation (middleware logs and operation system logs) with an easy operation. Executing FJQSS right after a problem occurs makes a rapid investigating of the causes.

Chapter 2 Collecting the information

This chapter explains the method of gathering the survey material, the confirm method of the survey material, and notes.

2.1 Collection method

This section explains the operating procedure of FJQSS (Information Collection Tool).

Collecting information can do by using the command. You can operate the Collection by the method that shows the product selection menu, or the method that does not show such the menu.

- 2.1.1 Collecting with showing menu
- 2.1.2 Collecting without showing menu

2.1.1 Collecting with showing menu

This subsection explains the operating procedure of FJQSS (Information Collection Tool) with showing the product selection menu.

Authority to execute

User Authority of the root user is necessary.

Prior confirmation

Set the output directory of the collected information if necessary. The output directory of default is the /var/tmp directory on Solaris, or the /tmp directory on Linux. See [2.3 Changing a path of the output directory] for details of changing the output directory.

Synopsis

If you execute the command specifying the output directory by the parameter:

```
fjqss_collect output_directory
```

If you execute the command specifying the output directory by the environment variable or the environmental configuration file:

```
fjqss_collect
```

Path to this command

```
/opt/FJSVqstl
```

Notes in executing the command

- Do not put fjqss_collect or its copy into any other directory. They work only if they are in the path shown above in [Path to this command].
- Execute this command immediately after the problem occurs.
- Multiple execution of this command is not available. If you do that, the following error message appears:

```
FJQSS is already started. Please restart FJQSS after the executing one ends.
```

Operating procedure

This part explains the operating procedure of FJQSS (Information Collection Tool).

1. Log on to the server, open the console window and execute the information collection command in the console window.

   Hereafter, it explains by the example when outputting the information to the default output directory:

```
> cd /opt/FJSVqstl
> ./fjqss_collect
```

2. The product selection menu appears. It shows the list of the products whose information can be collected.
3. Put the number of the product of which you want to collect the information:

```
-----------------------------------------------------
FJQSS Product Selection
-----------------------------------------------------
  1.MW Product B (Server Side)
  2.MW Product B (Client Side)

Please input a number of the product
of which you want to collect the information.>>1
```

4. Check the indicated name(s) of the middleware (s) whose investigation information are to be collected, then press the [Y] key to start.

   Collection starts and the following indicator appear in the command prompt:

```
The following product information is collected.
        MW Product B (Server Side)

Input 'Y-key' to collect data, or 'N-key' to stop.(default=N):y
preparing:            ######################################## 100% [00:01]
```

   The time and the disk space required for the collection depend on the condition of the target system.

5. Check the contents of the output directory.

   After the FJQSS (Information Collection Tool) has completed the collection, the following prompt appears. Verify that the information have been collected in the directory whose name is indicated in the prompt:

```
The following product information is collected.
        MW Product B (Server Side)

Input 'Y-key' to collect data, or 'N-key' to stop.(default=N):y
preparing:            ######################################## 100% [00:01]
collecting:           ######################################## 100% [00:01]
archiving:            ######################################## 100% [00:01]
compress:             ######################################## 100% [00:15]
Completed.
Gathered investigation informations archived to
    /tmp/ezcollect20121018172939/result20121018172939.tar.gz(206KB)
```

   This is an example on Linux.

   The gathered investigation informations are archived to /var/tmp/⋯ directory on Solaris, or /tmp/⋯ directory in Linux.

   The following files are created in the output directory (ezcollectYYYYMMDDHHMMSS: YYYYMMDDHHMMSS indicates the time (year, month, day, hours, minutes and seconds) that the collection started). See [2.2 Checking the collected information] for the output directory:

- resultYYYYMMDDHHMMSS.tar.gz

  The collected information.

- result.txt

  The result log is a text file to indicate if the collection of information has succeeded.

  This file indicates the names and the results of collection (OK/NG) of each material.

- ez.log

  The activity log of the FJQSS (Information Collection Tool).

Aborting FJQSS (Information Collection Tool)

This subsection explains how to abort running FJQSS (Information Collection Tool).

Although it is recommended that you wait until the collection ends, you can press [CTRL] + [C] key combination to abort the execution if it is necessary. The following prompt appears if [CTRL] + [C] is pressed and the execution is aborted.

```
The following product information is collected.
        MW Product B (Server Side)

Input 'Y-key' to collect data, or 'N-key' to stop.(default=N):y
preparing:            ################## 40% [00:01]
Stopped FJQSS process.
```

The output directory that has been made in the collecting might remain when the executing is aborted. There is no influence in the operation of FJQSS because the output directory will be newly made when FJQSS is executed again.

Moreover, there is no influence in the operation of FJQSS even if the remaining output directory was deleted.

Note

Do not close the command prompt before the collection ends. if you do so, the temporary directories might remain in the work directory. Please delete the following directories if they remain in the work directory.

・ezcollectYYYYMMDDHHMMSS

・ezworkYYYYMMDDHHMMSS

See 2.3 Changing a path of the output directory] for details of the work directory.

2.1.2 Collecting without showing menu

This subsection explains the operating procedure of FJQSS (Information Collection Tool) without showing the product selection menu.

You can use the procedure in the shell script.

Authority to execute

User Authority of the root user is necessary.

Prior confirmation

Verify the product identification name of the product of which you collect the information by the product information showing command. See 2.1.3 Getting the product identification name] for details.

Set the output directory of the collected information if necessary. The output directory of default is the /var/tmp directory on Solaris, or the /tmp directory on Linux. See [2.3 Changing a path of the output directory] for details of changing the output directory.

Synopsis

The specification of product identification name by "-pr <product identification name>" is necessary if you specify the silent mode option by "-s".

If you execute the command specifying the output directory by the parameter:

```
fjqss_collect -pr <product identification name> [-s] output_directory
```

If you execute the command specifying the output directory by the environment variable or the environmental configuration file:

```
fjqss_collect -pr <product identification name> [-s]
```

Description

The fjqss_collect command collects the information required for the investigation.

Option

-pr

This option specifies the product identification name of the product of which you collect the information.

-s

The command runs without asking to the operator. (It runs in silent mode.)

Path to this command

```
/opt/FJSVqstl
```

Notes in executing the command

- Do not put fjqss_collect or its copy into any other directory. They work only if they are in the path shown above in [Path to this command].
- Execute this command immediately after the problem occurs.
- Multiple execution of this command is not available. If you do that, the following error message appears:

```
FJQSS is already started. Please restart FJQSS after the executing one ends.
```

2.1.3 Getting the product identification name

This subsection explains how to get the product identification name which you specify when collecting the information without showing Product Selection Menu.

The product identification name can be got by executing the product information showing command.

Authority to execute

User Authority of the root user is necessary.

Synopsis

```
fjqss_mwpinfolist
```

Description

This command shows the list of the information of products of which the information collection is available.

Each product's information is shown in one line.

The product's Information is shown in order of "No." (number), "KEY" (product identification name), "DispName" (product name) and "Path" (destination directory of the information)

Example of the result of executing the product information showing command:

```
MW Product List
No. KEY        DispName                Path
--------------------------------------------------------------------------------
1.  PRODUCTA  MW PRODUCT A            /opt/FJSVPRDA/server/fjqss_tool1
2.  PRODUCTB  MW PRODUCT B            /opt/FJSVPRDB/server/fjqss_tool1
3.  PRODUCTC  MW PRODUCT C            /opt/FJSVPRDC/server/fjqss_tool1
```

If you collect the information of the product No.1, specify the product identification name "PRODUCTA" as the parameter

to collect the information of the "MW PRODUCT A" by the procedure of 2.1.2 Collecting without showing menu].

Path to this command

```
/opt/FJSVqstl
```

2.2 Checking the collected information

The collected information are created in the output directory (ezcollectYYYYMMDDHHMMSS) as file "resultYYYYMMDDHHMMSS.tar.gz".

The output directory is created at the output specified for an environmental configuration file destination at the output destination. If no destination is specified, then the output directory is created at the /var/tmp directory on Solaris, or the /tmp directory on LInux.

You can specify a path of the output directory if you execute the information collecting command in the console window. See [2.3 Changing a path of the output directory].for details of the environmental configuration file ant the output destination.

Verify that the following file and directory exist:

- Result log (result.txt)

  The result log is a text file to indicate if the collection of information has succeeded.

  This file indicates the names and the results of collection (OK/NG) of each material.

- Collected information (resultYYYYMMDDHHMMSS.tar.gz)

Note

Collected information (resultYYYYMMDDHHMMSS.tar.gz) is compressed and archived by gzip and tar. Collected information contains the following files.

- resultdir

  This directory contains the collected files.

- result.txt

  The result log is a text file to indicate if the collection of information has succeeded.

  This file indicates the names and the results of collection (OK/NG) of each material.

- ez.log

  The activity log of the FJQSS (Information Collection Tool).

- convert.txt

  If the file path name of information collecting is too long and information collecting can't get in former file name, file path name is converted and gathered. Former file path name and Converted file path name are output to convert.txt.convert.txt is output in the form of the following.

  Output form

  <Former file path name>,<Converted file path name>

  convert.txt is output when there was only information collecting for which the conversion of the file path name was necessary.

2.3 Changing a path of the output directory

You can change a path of the output directory by two methods below.

- Specifying a path of the output directory in the option of the information collecting command.
- Specifying a path of the output directory by setting the environment variable.
- Specifying a path of the output directory by updating the environmental configuration file.

The method of specifying in the option of the command explains in 2.1.1 Collecting with showing menu and 2.1.2 Collecting without showing menu]. Hereafter, the method of specifying by the environment variable and by the environmental configuration file will be explained.

-

Specifying a path of the output directory by setting the environment variable

You can specify the directory where the output directory will be created by setting the environment variable EZ_OUTPUTDIR before executing this command.

Description format:

```
EZ_OUTPUTDIR="the directory path where the output directory will be created"
export EZ_OUTPUTDIR
```

Example:

If “/etc/work” is specified to %EZ_OUTPUTDIR%:

```
EZ_OUTPUTDIR=/etc/work
export EZ_OUTPUTDIR
```

Then the directory below will be created as the output directory:

```
/etc/work/ezcollectYYYYMMDDHHMMSS
```

Specifying a path of the output directory by updating the environmental configuration file

Environmental configuration file (FJQSSconf.txt) is a file that defines the operating environment of FJQSS.

It exists in following directory.

```
<Directory where the product is installed>/fjqss_tooln (n=1,2,..,9)
```

The items that can be specified for environmental configuration file are shown below.

Item	Outline	Omission
OutputDir	The directory path where the output directory will be created (*3).	Possible (*2)
WorkDir	The directory path where the work directory (*1) will be created by FJQSS execution (*3).	Possible(*2)

(*1)The work directory is a directory that temporarily stores information necessary for the execution of FJQSS. When FJQSS ends, stored information is automatically deleted.

(*2) If the specification is wrong or omitted, the /var/tmp directory (on Solaris), or the /tmp directory (on Linux) is used.

(*3) You can specify the directory by the relative path from an absolute path or environmental configuration file.

The format of environmental configuration file is shown below.

Description format:

```
OutputDir="the directory path where the output directory will be created"
WorkDir=” work directory”
```

Example:

If “/etc/work” is specified to OutputDir, “/etc/tmp” is specified to WorkDIr.

```
OutputDir=/etc/work
WorkDir=/etc/tmp
```

Then the directory below will be created as the output directory:

```
/etc/work/ezcollectYYYYMMDDHHMMSS
```

And the directory below will be created as the work directory:

```
/etc/tmp/ezworkYYYMMDDHHMMSS
```

Note

If the plural specifications of the output directory is done, then the specification becomes effective in the following order by priority:

1. Specification in the option of the information collecting command.
2. Specification by setting the environment variable.
3. Specification by updating the environmental configuration file.

Appendix A Messages and error codes

A.1 Error Messages

The message which FJQSS outputs can have an error code. See [A.2 Error Codes]. for detail of error codes.

Look for	What to do
FJQSS is already started. Please restart FJQSS after the executing one ends.	Check if FJQSS is running. Multiple execution of it is not available.
The file used with FJQSS is not found. Please contact a Fujitsu technological member. (product= PRODUCT_IDETIFICATION_NAME, errcode= ERROR_CODE)	Please contact the Fujitsu engineer, and pass on the PRODUCT_IDENTIFICATION_NAME, PATH_NAME and ERROR_CODE in the message. About detail meanings of the ERROR_CODE, see [A.2 Error Codes].
Failed in the copy of materials used with FJQSS to the work directory. Please contact a Fujitsu technological member. (PATH= PATH_NAME, errcode= ERROR_CODE)	
Failed in the access of the file used with FJQSS. Please contact a Fujitsu technological member. (errcode= ERROR_CODE)	
The file used with FJQSS is not found or broken. Please contact a Fujitsu technological member. (PATH= PATH_NAME)	
Failed in the access of the directory used with FJQSS. Please contact a Fujitsu technological member. (PATH= PATH_NAME)	
Failed in the current directory change to the work directory. Please contact a Fujitsu technological member. (PATH= PATH_NAME)	
Failed to execute FJQSS.Please contact Fujitsu technical staff.	
There is a wrong option in the command line. Please restart FJQSS with the correct option.	Execute FJQSS again after removing the cause according to the left messages. About Environmental configuration file, see [2.3 Changing a path of the output directory].
Not super-user. Please restart FJQSS by super-user	
A memory is insufficient. Please stop unnecessary applications and restart FJQSS.	
The product specified by the -pr option is not found. Please restart FJQSS with the correct option. (product= PRODUCT_IDETIFICATION_NAME)	
Japanese characters are included in the directory name at the 'OutputDir' of environmental file. Japanese characters cannot be used. Please correct environmental file and restart FJQSS. (environmental file=PATH_NAME)	

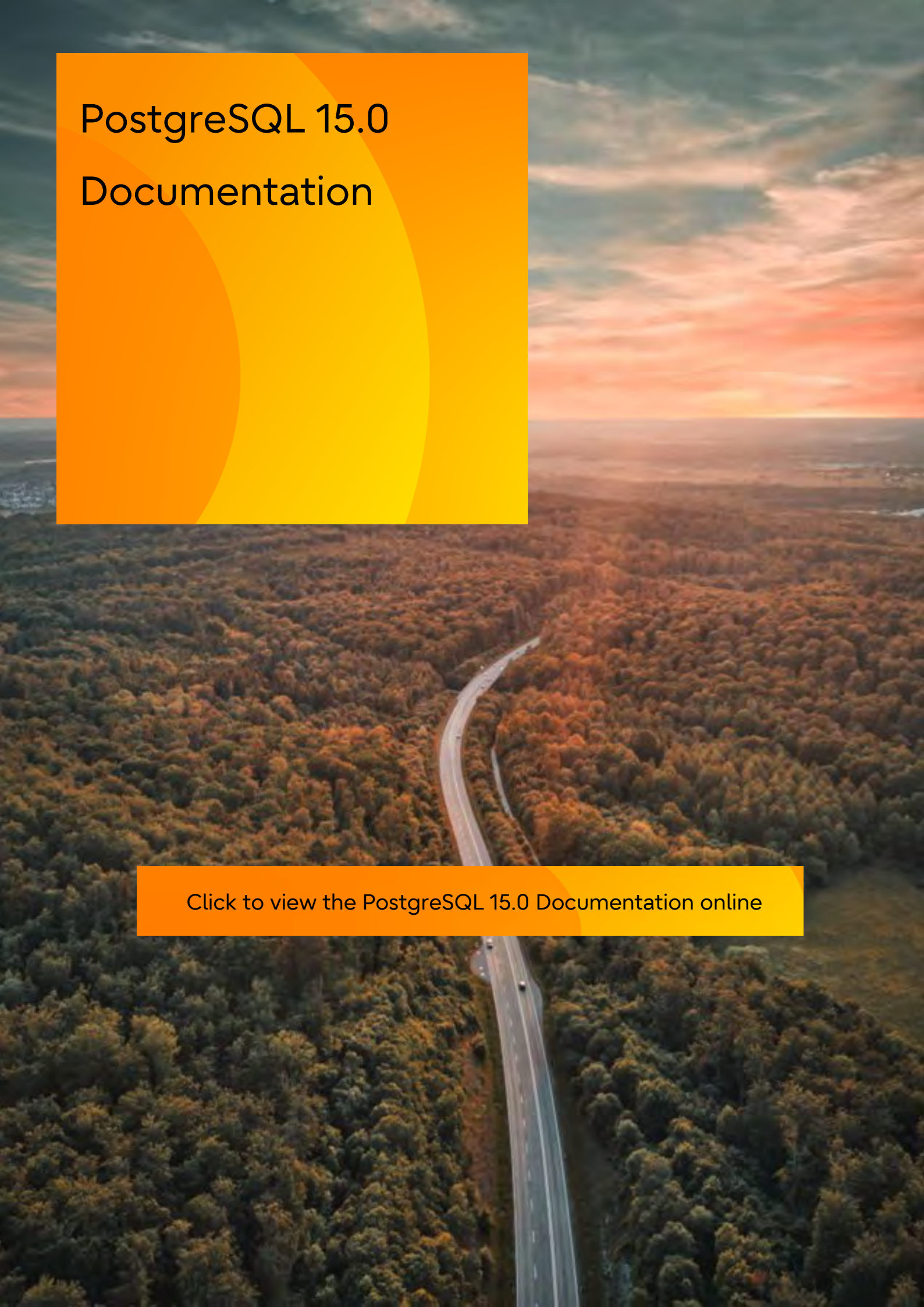
Look for	What to do
Japanese characters are included in the directory name at the 'WorkDir' of environmental file. Japanese characters cannot be used. Please correct environmental file and restart FJQSS. (environmental file=PATH_NAME)	
Failed in making the output directory. Please confirm the following. -There must be an authority to make the folder. -Passing in xx bytes or less must be specified. -The length of each directory name must be 255 bytes or less.	
Failed in making the work directory. Please confirm the following. -There must be an authority to make the folder. -Passing in xx bytes or less must be specified. -The length of each directory name must be 255 bytes or less.	
Output directory is too long at the 'OutputDir' of environmental file.Please adjust the output directory to xx bytes or less and restart FJQSS (environmental file= PATH_NAME)	
Work directory is too long at the 'WorkDir' of environmental file. Please adjust the output directory to xx bytes or less and restart FJQSS (environmental file= PATH_NAME)	
The output directory is too long specified in the option. Please adjust the output directory to %s bytes or less and restart FJQSS.	
Japanese characters are included in the output directory specified in the option. Japanese characters cannot be used. Please correct the output directory and restart FJQSS.	
The output directory is too long specified with the environment variable:EZ_OUTPUTDIR. Please adjust environment variable:EZ_OUTPUTDIR to xx byte or less.	
Failed to create log.(PATH=PATH_NAME) Take action indicated the following items: - Create the output folder if it doesn't exist. - Set the role to the output folder.	
Products are not installed.Please make sure that the products that are installed.	
The is no installed product of which FJQSS can collect the information. FJQSS ends.	
Stopped FJQSS process.	No user action is necessary.

A.2 Error Codes

Error Code	Cause
0	No error.

Error Code	Cause
1	Command operand error
2	Insufficient memory
3	Failed to execute without administrator permissions
5	Though "-s" option exists, "-pr" option does not exist
6	FJQSS common component has been already done
10	Middleware products' list file does not exist
11	Middleware product specified with -pr option does not exist
13	The built-in directory of the target middleware product of collecting information does not exist
15	Japanese characters are included in the directory name at the output directory of environmental file
16	Japanese characters are included in the directory name at the work directory of environmental file
17	Too long path name is specified in the directory name at the output directory of environmental file
18	Too long path name is specified in the directory name at the work directory of environmental file
19	Failed to create the output directory
20	Failed to create the work directory
21	Failed to access the directory
22	Failed to copy the common materials and built-in materials of the product directory to the work directory
23	Failed to show the information of the product to collect its information materials
24	Failed to change the current directory to the work directory
25	Too long path name is specified to the parameter that specifies the output directory
26	Japanese characters are included in the parameter that specifies the output directory
27	Products of which collecting information is available are not installed
28	Failed to execute FJQSS

PostgreSQL 15.0 Documentation



Click to view the PostgreSQL 15.0 Documentation online