

FUJITSU Enterprise Postgres 9.5



Cluster Operation Guide

Linux

J2UL-2147-01ENZO(00)
June 2016

Preface

Purpose of This Document

This document explains the items required to operate FUJITSU Enterprise Postgres on a cluster system.

Intended Readers

This document is aimed at people who install and operate FUJITSU Enterprise Postgres on a cluster system. 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:

Part 1 Overview of Cluster Systems

Chapter 1 Definition of Cluster Operation

Provides an overview of cluster operations and their types.

Part 2 Database Multiplexing Mode

Chapter 2 Overview of Database Multiplexing Mode

Provides an overview of database multiplexing mode.

Chapter 3 Setting up Database Multiplexing Mode

Describes how to set up database multiplexing mode.

Chapter 4 Operations in Database Multiplexing Mode

Explains periodic database multiplexing mode.

Chapter 5 Action Required when an Error Occurs in Database Multiplexing Mode

Explains the action required when an error occurs during a database multiplexing mode.

Chapter 6 Managing Mirroring Controller Using WebAdmin

Explains how to setup 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.

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

First edition: June 2016

Copyright

Copyright 2016 FUJITSU LIMITED

Contents

Part 1 Overview of Cluster Systems.....	1
Chapter 1 Definition of Cluster Operation.....	2
1.1 What Is a Cluster System?.....	2
1.2 Cluster Operation Provided by FUJITSU Enterprise Postgres.....	2
Part 2 Database Multiplexing Mode.....	3
Chapter 2 Overview of Database Multiplexing Mode.....	4
2.1 What is Database Multiplexing Mode.....	4
2.2 System Configuration for Database Multiplexing Mode.....	6
2.2.1 Mirroring Controller Resources.....	7
2.2.2 Redundancy of the Admin and Log Transfer Networks.....	8
2.2.3 Notes on CPU Architecture and Products.....	9
2.2.4 Notes on Executing Reference Jobs on the Standby Server.....	9
2.3 Security in Database Multiplexing.....	9
2.3.1 Authentication of the Standby Server.....	10
2.3.2 Encryption of Transaction Logs Transferred to the Standby Server.....	11
Chapter 3 Setting up Database Multiplexing Mode.....	12
3.1 Installation.....	13
3.2 Preparing the Backup Disk.....	13
3.3 Setting up the Primary Server.....	13
3.3.1 Setting up Database Multiplexing Mode on the Primary Server.....	13
3.3.2 Creating, Setting, and Registering the Primary Server Instance.....	15
3.4 Setting up the Standby Server.....	18
3.4.1 Setting up Database Multiplexing Mode on the Standby Server.....	18
3.4.2 Creating, Setting, and Registering the Standby Server Instance.....	19
3.5 Confirming the Streaming Replication Status.....	21
3.6 Creating Applications.....	22
3.6.1 Application Connection Server Settings.....	22
3.6.2 Notes on Indexes.....	22
3.7 Checking the Behavior.....	22
3.8 Tuning.....	22
3.8.1 Tuning to Stabilize the Database Multiplexing Mode.....	22
3.8.2 Tuning to Stabilize Queries on the Standby Server.....	23
3.8.3 Tuning to Stabilize Queries on the Standby Server (when Performing Frequent Updates on the Primary Server).....	23
3.8.4 Tuning for Optimization of Degrading Operation Using Abnormality Monitoring.....	23
3.9 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances.....	26
3.10 Backup Operation.....	28
3.10.1 Backing up Database Multiplexing Mode Information.....	28
3.10.2 Database Backup Operation.....	28
Chapter 4 Operations in Database Multiplexing Mode.....	30
4.1 Starting and Stopping Instances.....	30
4.2 Checking the Database Multiplexing Mode Status.....	30
4.3 Manually Switching the Primary Server.....	31
4.4 Manually Disconnecting the Standby Server.....	31
4.5 Monitoring Mirroring Controller Messages.....	32
4.6 Server Maintenance.....	33
4.6.1 Rolling Updates.....	33
4.6.2 Stopping for Maintenance.....	38
4.7 Changes in Operation.....	38
4.7.1 Changes Required when the Standby Server is Stopped.....	39
4.7.2 Changing from Single Server Mode to Database Multiplexing Mode.....	39
4.7.3 Changing from Database Multiplexing Mode to Single Server Mode.....	40

Chapter 5 Action Required when an Error Occurs in Database Multiplexing Mode.....	42
5.1 Action Required when Server Degradation Occurs.....	42
5.1.1 Operations when the Server has Started Degrading after a Switch has Occurred.....	42
5.1.1.1 Identify Cause of Error and Restore the Standby Server.....	44
5.1.1.1.1 Stop Mirroring Controller.....	44
5.1.1.1.2 Recovery of the Mirroring Controller management directory.....	45
5.1.1.1.3 Identify cause of error and perform recovery.....	45
5.1.1.2 Rebuild the Standby Server.....	46
5.1.1.3 Failback of the Primary Server.....	46
5.1.2 Operations when the Server has Started Degrading after a Disconnection has Occurred.....	47
5.1.2.1 Identify Cause of Error and Restore the Standby Server.....	48
5.1.2.1.1 Stop Mirroring Controller.....	48
5.1.2.1.2 Recovery of the Mirroring Controller management directory.....	49
5.1.2.1.3 Identify cause of error and perform recovery.....	49
5.1.2.2 Rebuild the Standby Server.....	49
5.1.3 Addressing Errors During Degrading Operation.....	49
5.2 Action Required when Automatic Switch Fails.....	50
5.3 Action Required when Automatic Disconnection Fails.....	51
5.4 Action Required when All Servers or Instances Stopped.....	51
5.5 Recovering from an Incorrect User Operation.....	55
Chapter 6 Managing Mirroring Controller Using WebAdmin.....	57
6.1 Mirroring Controller Setup.....	57
6.2 Stopping Mirroring Controller.....	59
6.3 Starting Mirroring Controller.....	59
6.4 Disabling Failover Mode.....	60
6.5 Enabling Failover Mode.....	60
6.6 Deleting Mirroring Controller Setup.....	60
6.7 Status Update after Failover.....	61
6.8 Action Required when an Error Occurs in the Combined Admin Network and Log Transfer Network.....	62
Appendix A Parameters.....	63
A.1 Parameters Set on the Primary Server.....	63
A.2 Parameters Set on the Standby Server.....	64
A.3 Network Configuration File.....	65
A.4 Server Configuration File.....	66
Appendix B Supplementary Information on Building the Primary Server and Standby Server on the Same Server.....	68
B.1 Backup Data Storage Destination Directory.....	68
B.2 How to Execute the mc_ctl Command.....	68
Index.....	70

Part 1 Overview of Cluster Systems

This part provides an overview of the cluster systems, and describes the types and operation modes.

Chapter 1 Definition of Cluster Operation.....	2
--	-------------------

Chapter 1 Definition of Cluster Operation

This chapter explains the features of a cluster system, the types of cluster operations, and the operation mode.

1.1 What Is a Cluster System?

A cluster system is a system that joins multiple machines (hereafter, referred to as servers) to achieve high availability and scalability throughout the system.

This allows the creation of systems that simultaneously achieve high availability and scalability.

High availability

For example, by transferring the role to another server when a server in the cluster system fails, the downtime for the entire system can be made shorter.

This type of operation mode is referred to as a "failover operation".

Scalability

The load should be distributed, in order to extend the processing capacity across the entire system by dividing multiple application processes amongst multiple servers, regardless of processing content, or to suit the processing content.

1.2 Cluster Operation Provided by FUJITSU Enterprise Postgres

In FUJITSU Enterprise Postgres, the following failover operation mode is provided in each system:

- Log shipping mode

Log shipping mode

This operation mode is 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 sending and reflecting the updated transaction log of the database from the server that receives the update (primary server) to another server (standby server).

In addition, the client driver automatically distinguishes between the primary and standby servers, so applications can be connected transparently regardless of the physical server. Accordingly, a flexible system can easily be built, for example a system that offloads connections to the standby server if the connections only perform referencing.



FUJITSU Enterprise Postgres supports only configurations comprising one standby server per primary server.



- 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.

- Refer to "[Part 2 Database Multiplexing Mode](#)" for information on the database multiplexing feature.

Part 2 Database Multiplexing Mode

This part describes database multiplexing mode.

Chapter 2 Overview of Database Multiplexing Mode.....	4
Chapter 3 Setting up Database Multiplexing Mode.....	12
Chapter 4 Operations in Database Multiplexing Mode.....	30
Chapter 5 Action Required when an Error Occurs in Database Multiplexing Mode.....	42
Chapter 6 Managing Mirroring Controller Using WebAdmin.....	57

Chapter 2 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.

2.1 What is Database Multiplexing Mode

This operation mode is 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.

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. The database will be copied in synchronous mode.

Monitoring using database multiplexing mode

- 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.

- 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. A disk error is detected when an I/O error occurs.

Failures that can be detected are those that physically affect the entire system, such as disk header or device power failures.

Figure 2.1 Failover from the primary server to the standby server

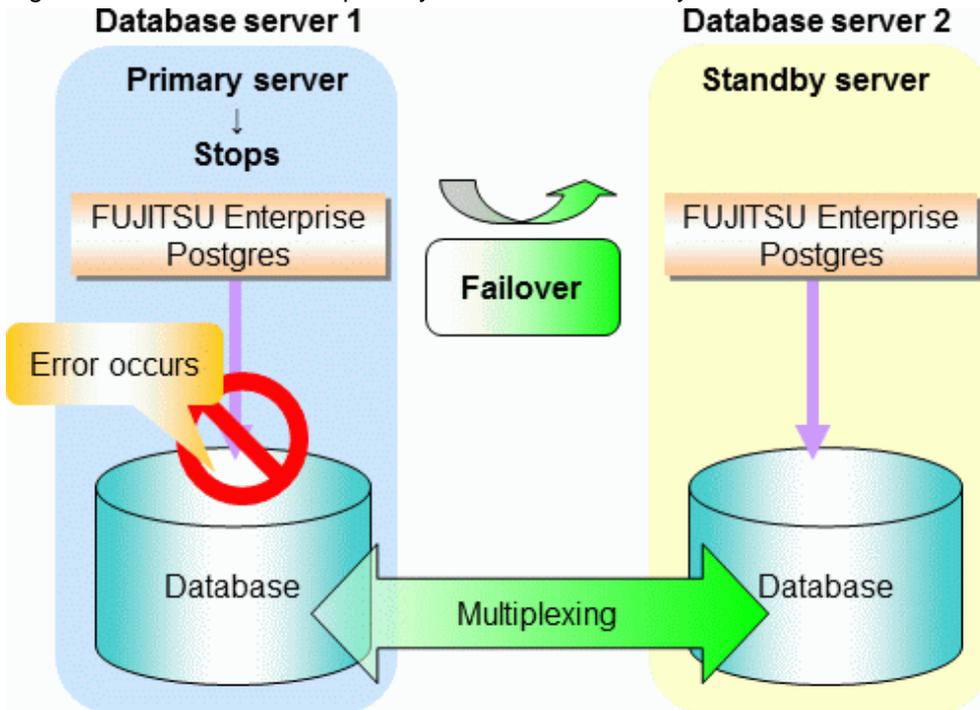
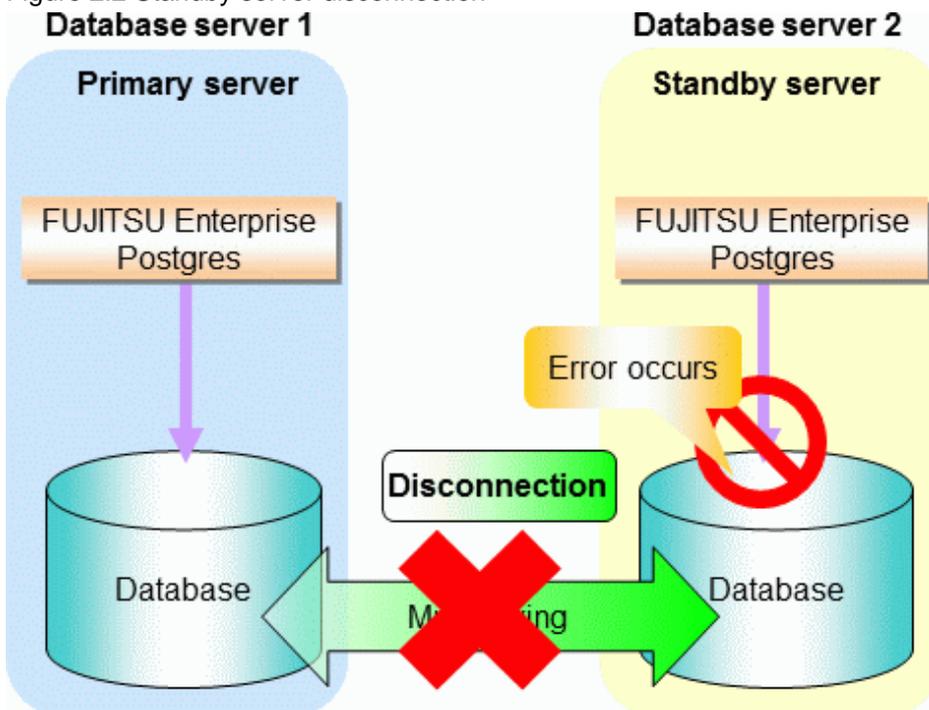


Figure 2.2 Standby server disconnection



When database multiplexing is used to perform monitoring and an error is detected, this option will automatically switch the primary server or disconnect the standby server.

Reference jobs can also be executed on the standby server.

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 "[5.1 Action Required when Server Degradation Occurs](#)" for details.

See

- Refer to "[2.2.4 Notes on Executing Reference Jobs on the Standby Server](#)" for information on reference jobs.
- The features that can be used will depend on whether automatic switch/disconnection is used. Refer to "Features that can be Installed" in the Installation and Setup Guide for Server for details.

Point

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.

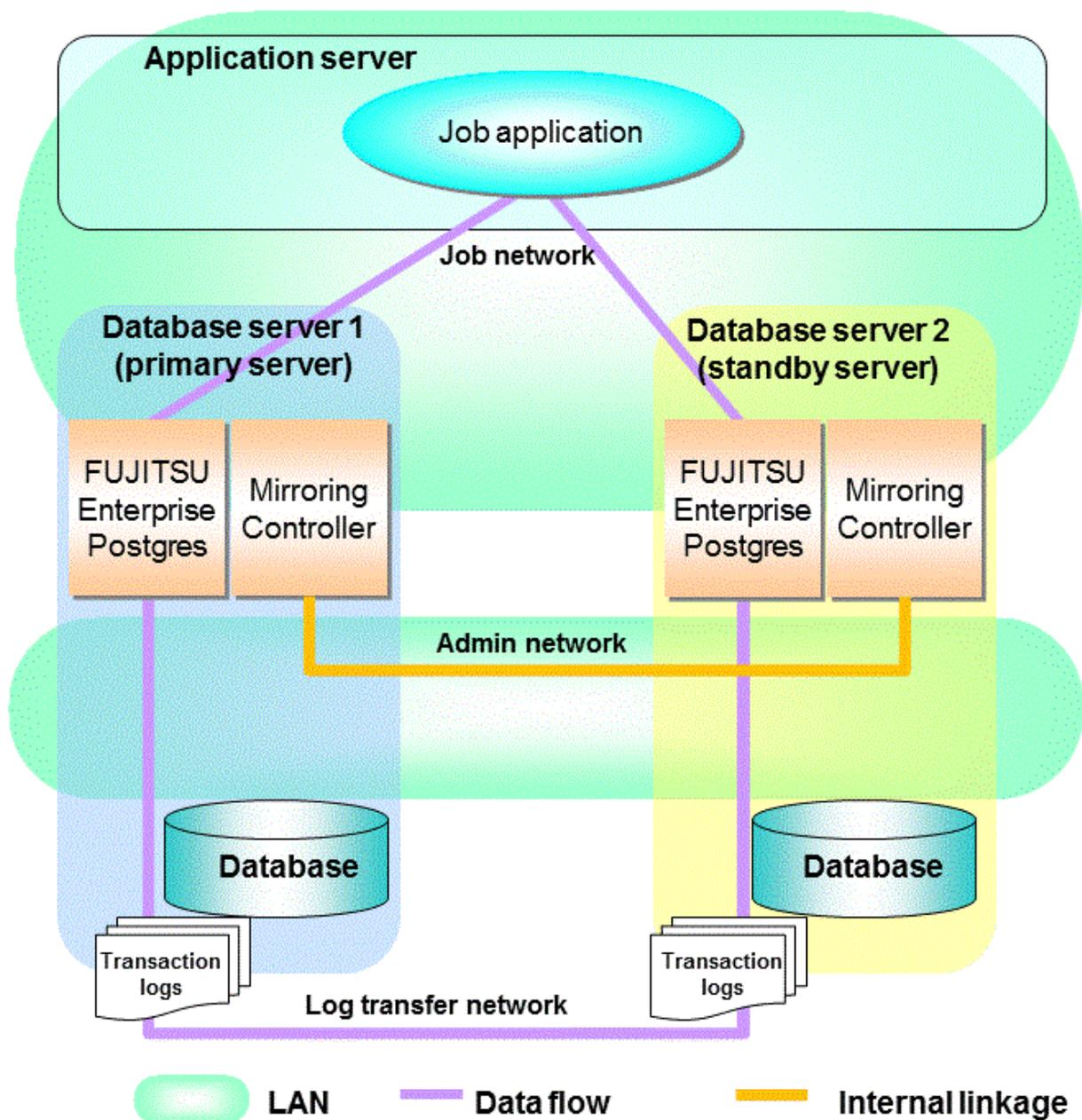
2.2 System Configuration for Database Multiplexing Mode

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.
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 2.3 System configuration for database multiplexing mode



2.2.1 Mirroring Controller 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

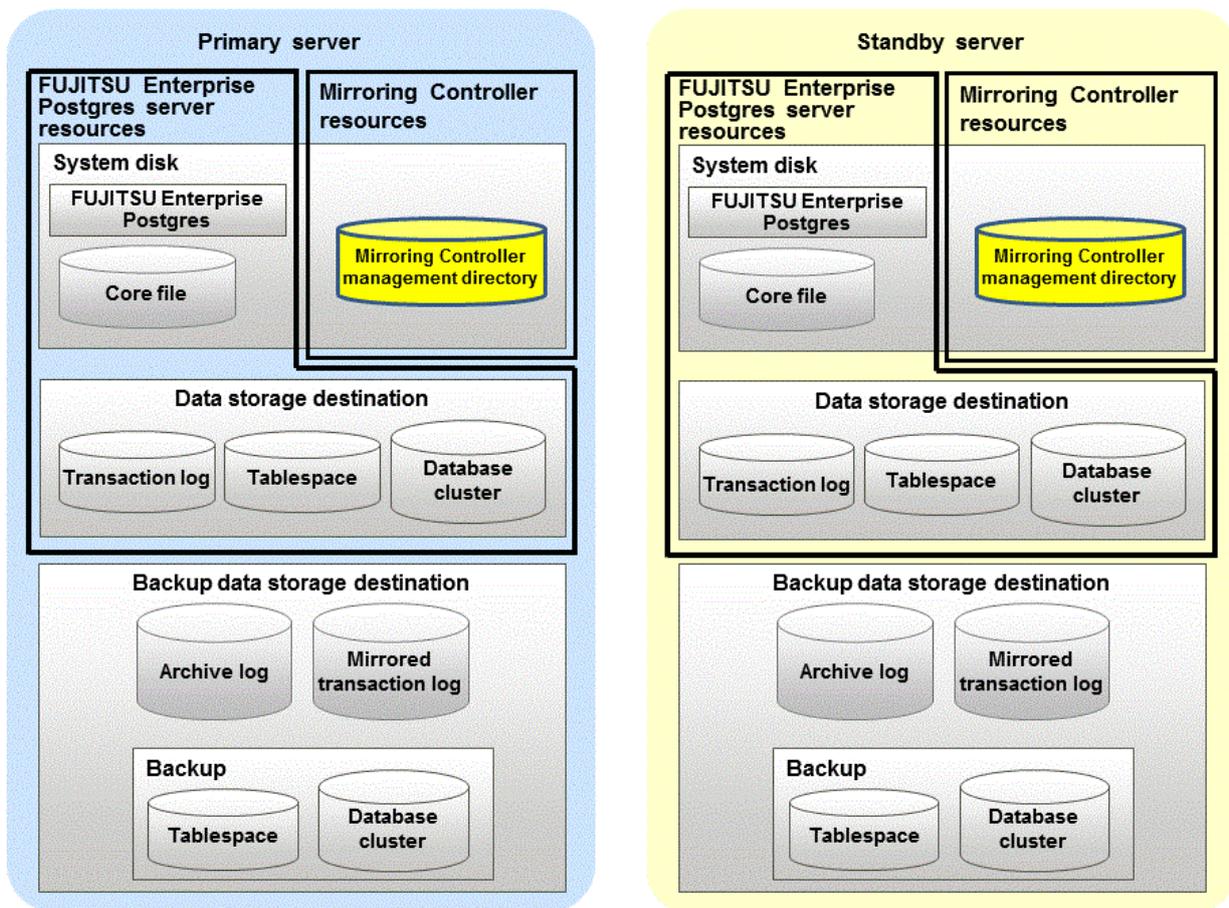
- Do not create the Mirroring Controller management directory in a directory managed by FUJITSU Enterprise Postgres, otherwise it may be deleted by mistake or may cause unexpected problems when FUJITSU Enterprise Postgres recovery is performed (such as old version of files being restored).

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.
- If the automatic switch/disconnection is enabled, do not edit synchronous_standby_names for the Mirroring Controller monitoring target instance using the methods below. Otherwise, disconnection by the Mirroring Controller may fail, since postgresql.conf settings are overridden by values specified in postgresql.auto.conf and the include_dir directives.
 - Setting the parameter using the ALTER SYSTEM statement
 - Editing the value specified in postgresql.conf or in a configuration file located in a directory listed in the include_dir directive

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 2.4 Configuration when backing up Mirroring Controller resources



2.2.2 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.

2.2.3 Notes on CPU Architecture and Products

Use the same CPU architecture (endian) for the primary server and the standby server. Additionally, use the same bit architecture (32-bit/64-bit) for the FUJITSU Enterprise Postgres installation.

A server using only PostgreSQL streaming replication cannot be specified as the database multiplexing system log transfer destination.

2.2.4 Notes on Executing Reference Jobs on the Standby Server

In database multiplexing mode where automatic switch/disconnection is enabled, switch and disconnection are performed automatically. If a switch occurs when executing a reference job for the standby server, it may affect the performance of that job. The reason is that, 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 executed on the original standby server will be processed. Therefore, before executing a job using this configuration, give careful consideration to the server resource estimates, and the likely impact on performance.

Accordingly, if you are concerned about the impact on performance, pause the reference job on the original standby server, change the original primary server to the new standby server, and then resume the reference job for the new standby server.

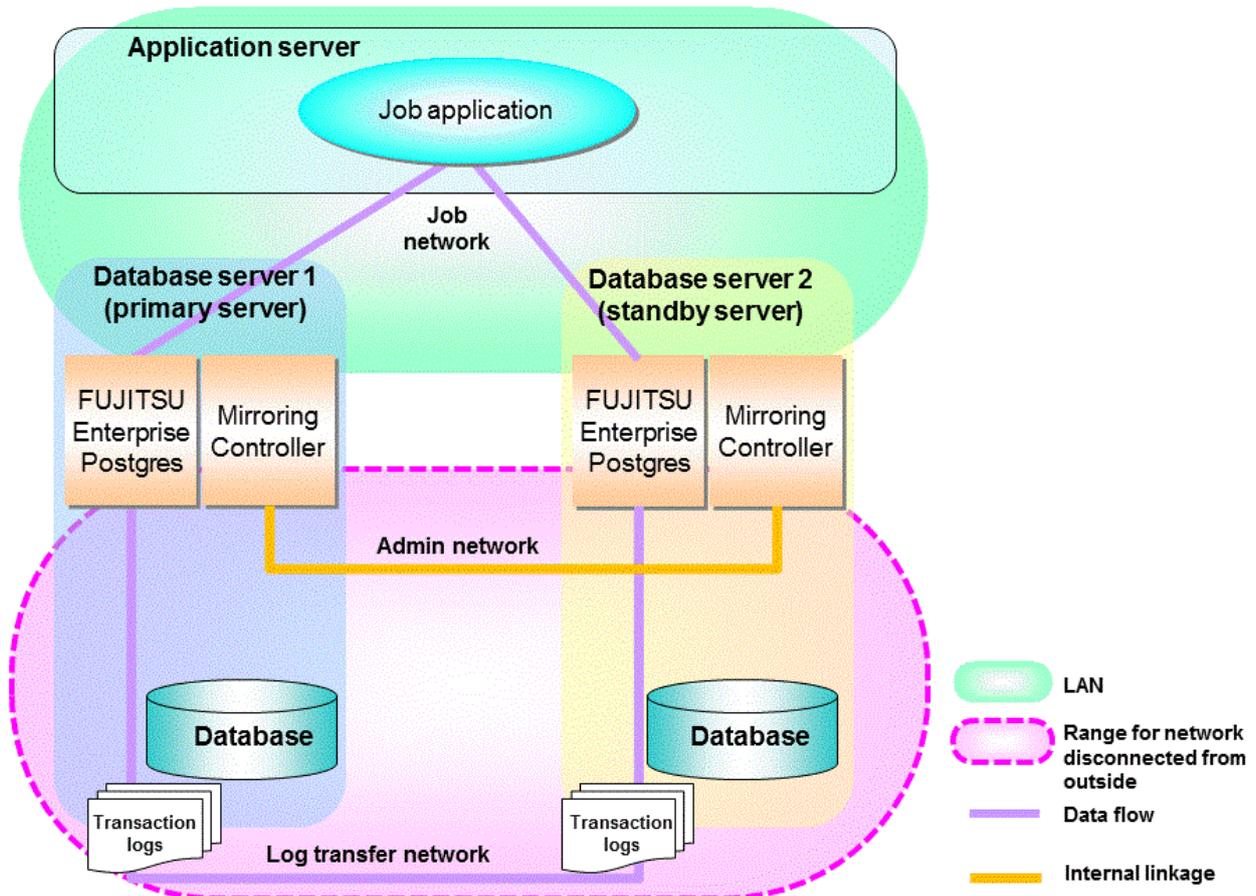
2.3 Security in Database Multiplexing

Database multiplexing mode 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 protect the confidentiality of data when transferring the transaction logs, carefully consider security and take note of the following:

- 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 below.

Figure 2.5 Security



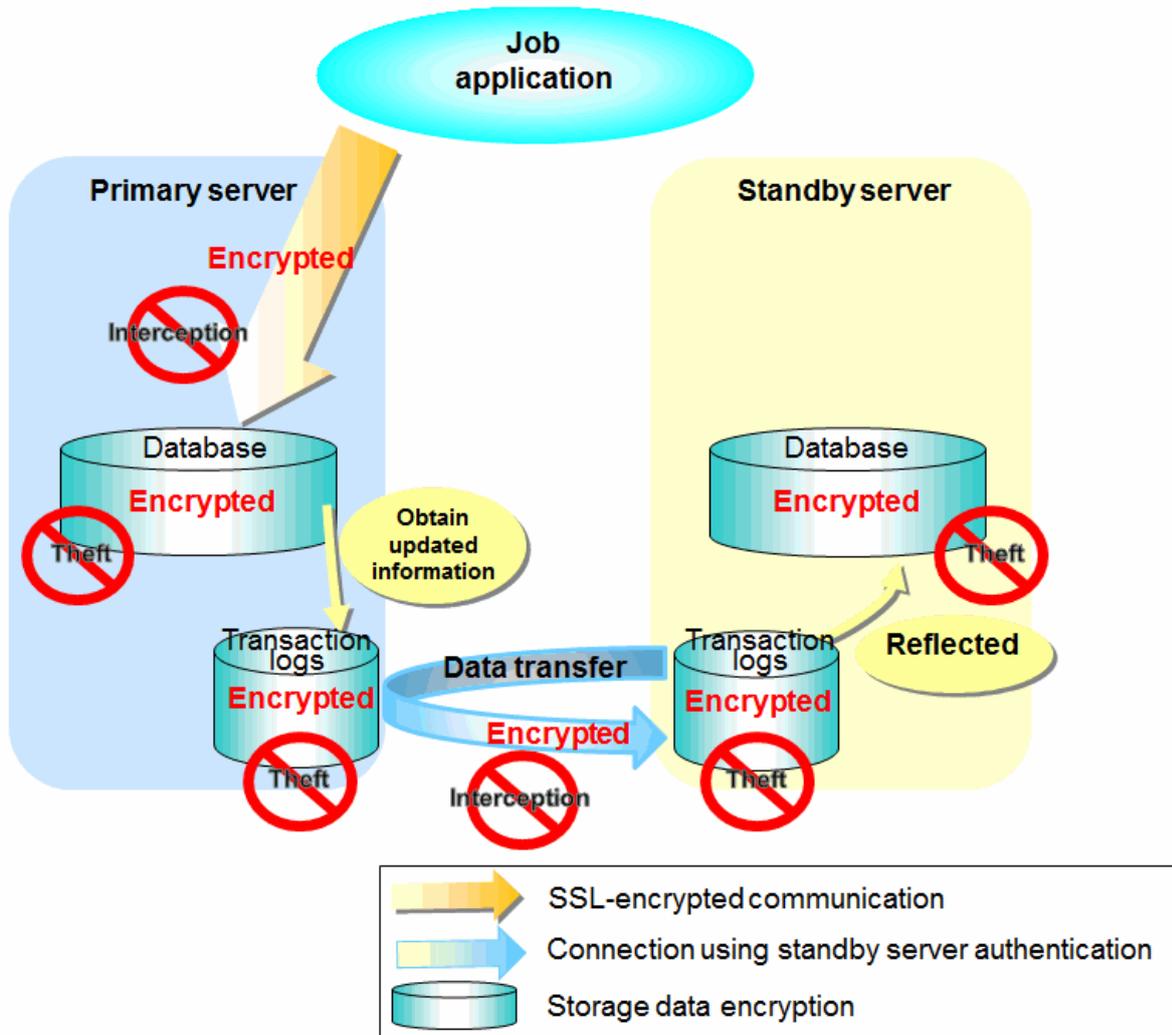
However, it may not always be possible to adopt the configuration mentioned above. For example, you may want to place the servers in a nearby/neighboring office to minimize network delays.

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.

Figure 2.6 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.

2.3.1 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`.
.....

2.3.2 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 3 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

Setup and operations of the database multiplexing mode must be performed by the instance administrator user.

The instance administrator user must also be registered as a database superuser.



Note

When executing the `initdb` command, specify the `-U` option, and only perform operations in which the instance administrator user (who is the command execution user) is also a database superuser.

If an operation in which the instance administrator user is not a database superuser is performed, Mirroring Controller will be unable to connect to the instance, and therefore will not be able to perform error monitoring for the database multiplexing mode.

Matching the system times

Before starting the setup, ensure that the times in both servers 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.

Setup

Perform the procedure in the table below to set up database multiplexing mode.

Database multiplexing mode uses the streaming replication feature. If you know how to set up streaming replication, the setup of database multiplexing mode will be easy.

Step	Task		Refer to
	Primary server	Standby server	
1	Installation		3.1 Installation
2	Preparing the backup disk		3.2 Preparing the Backup Disk
3	Setting up database multiplexing mode		3.3.1 Setting up Database Multiplexing Mode on the Primary Server
4	Creating, setting, and registering the instance		3.3.2 Creating, Setting, and Registering the Primary Server Instance
5		Setting up database multiplexing mode	3.4.1 Setting up Database Multiplexing Mode on the Standby Server
6		Creating, setting, and registering the instance	3.4.2 Creating, Setting, and Registering the Standby Server Instance
7	Confirming the streaming replication status		3.5 Confirming the Streaming Replication Status
8	Creating applications		3.6 Creating Applications
9	Checking the behavior		3.7 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 "[4.7.2 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](#)".

3.1 Installation

Refer to the manuals of each product, and then install the product.

See

- Refer to the Installation and Setup Guide for Server for details on how to install FUJITSU Enterprise Postgres.
-

3.2 Preparing the Backup Disk

In Mirroring Controller, backup allows recovery to be performed even if all server disks are corrupted.

The content on the primary server will be backed up. However, through switching and failback, the standby server may also become the primary server. Accordingly, prepare a backup disk device for the primary and standby servers. Ensure that backup is performed using the backup of the primary server.

3.3 Setting up the Primary Server

This section explains how to set up the primary server.

3.3.1 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 excluding spaces 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.conf file, 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. In the network.conf file, set "Read" permission 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.

Path of the sample file

```
/install/Dir/share/mc_network.conf.sample
```

In the network.conf file, specify the host name (or IP address) and port number of the primary server and standby server, and define the network configuration that will link between the Mirroring Controller processes.

Refer to "[A.3 Network Configuration File](#)" for details.

A definition example is shown below.

In this example, the Mirroring Controller default port number 27540 is specified, and "server1" and "server2" represent any server identifiers.

Example)

```
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.

In the *serverIdentifier.conf* file, set "Read" permission 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.

Path of the sample file

```
/install/Dir/share/mc_server.conf.sample
```

Set the parameters shown in the table below in the *serverIdentifier.conf* file.

Table 3.1 Parameters

Parameter	Content specified	Remarks
db_instance	<i>'dataStorageDestinationDir'</i>	Enclose the directory within single quotation marks ('). Use ASCII characters excluding spaces to specify this parameter.

Parameter	Content specified	Remarks
db_instance_password	'Password of the instance administrator user'	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 excluding spaces to specify this parameter. Enclose the name within single quotation marks (').
syslog_ident	' <i>programName</i> '	Specify the program name to be used to identify the Mirroring Controller messages in the system log. Use ASCII characters excluding spaces to specify this parameter. 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. Enclose the name within single quotation marks (').
heartbeat_interval	800 (milliseconds)	The time until an operation, such as the switch of the primary server, that is performed when there is no response, is calculated according to the following formula: $(\text{heartbeat_timeout (seconds)} + \text{heartbeat_interval (milliseconds)/1000}) \times (\text{heartbeat_retry (count)} + 1) = (1+800/1000) \times (2+1) = 5.4 \text{ (seconds)}$ According to this setting, errors are monitored at 800-millisecond intervals, and if there were three successive occurrences when there was no response for a duration of one second, the operation to switch the primary server, or disconnect the standby server from the primary server, is performed. When setting this parameter, there are some considerations to take into account to optimize degradation using abnormality monitoring. Refer to " 3.8.4 Tuning for Optimization of Degrading Operation Using Abnormality Monitoring " for details.
heartbeat_timeout	1 (second)	
heartbeat_retry	2 (count)	

3.3.2 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 excluding spaces 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 "5.1.1.1.3 Identify cause of error and perform recovery" for details.

2. Configure the encryption settings for the storage data.

Create the keystore file.

Refer to "Database Multiplexing Mode" 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	<i>standbyServerAddress</i>	<i>authenticationMethod</i>
	host	replication	fsep	<i>primaryServerAddress</i>	<i>authenticationMethod</i>

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.

As the authentication method, set a 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 OS user account of the instance administrator 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. 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.

As the authentication method, set a method other than trust authentication.

- 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.

An example of setting the authentication method is shown below.

#	TYPE	DATABASE	USER	ADDRESS	METHOD
	host	postgres	fsep	127.0.0.1/32	<i>authenticationMethod</i>

Refer to the note in step 3 for notes relating to trust authentication.

- 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 3.2 Parameters

Parameter	Content specified	Remarks
wal_level	Literal "hot_standby"	
max_wal_senders	Number of standby servers + 1	The number of standby servers (<i>n</i>) is 1, so 2 must be specified.
synchronous_standby_names	'standbyServerName'	Specify the name that will identify the standby server. (Enclose the name within single quotation marks (.)) Do not change this parameter while Mirroring Controller is running.
hot_standby	on	Specify this to execute reference jobs on the standby server. (*1)
wal_keep_segments	Number of file segments	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. Setting this parameter requires consideration regarding stabilization of the database multiplexing mode (refer to "3.8.1 Tuning to Stabilize the Database Multiplexing Mode" for details). Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on estimating the WAL segment.
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 of (heartbeat_interval + heartbeat_timeout/1000) x (heartbeat_retry + 1) of the <i>serverIdentifier.conf</i> file, 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 in the transaction log transfer on the standby server. By aligning this value with the value of (heartbeat_interval + heartbeat_timeout) x heartbeat_retry of the <i>serverIdentifier.conf</i> file, 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_xlogcopy.cmd "%p" " <i>backupDataStorageDestin</i>	Specify the command and storage destination to save the transaction log.

Parameter	Content specified	Remarks
	<i>ationDirectory/</i> archived_xlog/%f"	
backup_destination	Backup data storage destination directory	Specify the name of directory where to store the backup data. Set the privileges 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	The limit value for the number of standby server connections + number of simultaneous executions of mc_ctl status (*2) + 2 + 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.

*1: Mandatory to use the Mirroring Controller.

*2: Number of simultaneous executions of the mc_ctl command in the status mode.

6. Start the Mirroring Controller process.

Enabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode with the -f option specified. This 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 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode.

Example)

```
$ mc_ctl start -M /mcdir/inst1 -w
```

7. Obtain the backup.

Use the pgx_dmpall command to collect the backup.

3.4 Setting up the Standby Server

This section explains how to set up the standby server.

3.4.1 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_segments` 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_segments` parameter. Additionally, setting the `wal_keep_segments` parameter requires consideration regarding stabilization of the database multiplexing mode (refer to "[3.8.1 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 "`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 excluding spaces 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" permission 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" permission for the instance administrator user only. If users other than the instance administrator user are granted access privileges, the `mc_ctl` command will not work.

3.4.2 Creating, Setting, and Registering the Standby Server Instance

This section explains how to create, set, and register the standby server instance.



- 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. Configure the encryption settings for the storage data.
Deploy a copy of the keystore file of the primary server on the standby server.
Refer to "Database Multiplexing Mode" in the Operation Guide for details.
3. Execute the pg_basebackup command to create a copy of the primary server instance on the standby server. A recovery.conf file is also created using the connection information for the primary server specified in each option used with the mc_ctl command.

Example)

```
$ pg_basebackup -D /database/inst1 --xlog --progress --verbose -R --
dbname='application_name=standbyServerName' -h primaryServerIpAddress -p primaryServerPortNumber
```



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 recovery.conf file, enabling connections to be performed automatically.

If a password is not set in recovery.conf, 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.

4. Set the parameters shown in the table below in the postgresql.conf file.

Table 3.3 Parameters

Parameter	Content specified	Remarks
synchronous_standby_names	<i>'primaryServerName'</i>	Required after switching the primary server and then changing the original primary server to the new standby server. Enclose the name within single quotation marks ('). Do not change this parameter while Mirroring Controller is running.

5. Start the Mirroring Controller process.

After ensuring that the Mirroring Controller process of the primary server has started, start the Mirroring Controller process.

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 /mdir/inst1 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

```
$ mc_ctl start -M /mcdir/inst1 -w
```

6. 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
```

3.5 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                | 14336
usesysid           | 10
username           | fsep
application_name   | standby
client_addr        | 192.0.2.210
client_hostname    |
client_port        | 27500
backend_start      | 2015-04-23 20:24:13.761424+09
backend_xmin       |
state              | streaming
sent_location      | 0/3000060
write_location     | 0/3000060
flush_location     | 0/3000060
replay_location    | 0/3000060
sync_priority      | 1
sync_state         | sync
```

2. Confirm the search results of step 1.

Ensure that the connection established with the intended standby server is in synchronous mode.

Table 3.4 Items to be checked

Item	Required value
<code>application_name</code>	Value specified for <code>synchronous_standby_names</code> parameter in the <code>postgresql.conf</code> file of the primary server.
<code>client_addr</code>	IP address of the standby server.
<code>state</code>	"streaming".
<code>sync_state</code>	"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.

3.6 Creating Applications

This section explains how to create applications using database multiplexing, and points that should be noted when you create the applications.

3.6.1 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.

3.6.2 Notes on Indexes

Do not use hash indexes.

Even if a hash index is updated, a transaction log will not be output and the hash index will not be reflected on the standby server.

3.7 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.

3.8 Tuning

This section explains how to tune database multiplexing mode.

3.8.1 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 3.5 Parameters

Parameter	Content
<code>wal_keep_segments</code>	Refer to "3.3.2 Creating, Setting, and Registering the Primary Server Instance" for details.
<code>max_wal_size</code>	<p>The transaction log is written out according to the checkpoint trigger.</p> <p>If a transaction log with the capacity of the value specified in this parameter is generated, the checkpoint will be executed.</p> <p>If a large value is specified in this parameter, the time required for crash recovery will increase.</p> <p>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.</p>

3.8.2 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.
-

3.8.3 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.

Parameter	Description
<code>hot_standby_feedback</code>	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)
<code>vacuum_defer_cleanup_age</code>	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.



- Refer to "Standby Servers" in the PostgreSQL Documentation for details on the `hot_standby_feedback` parameter.
 - Refer to "Master Server" in the PostgreSQL Documentation for details on the `vacuum_defer_cleanup_age` parameter.
-

3.8.4 Tuning for Optimization of Degrading Operation Using Abnormality Monitoring

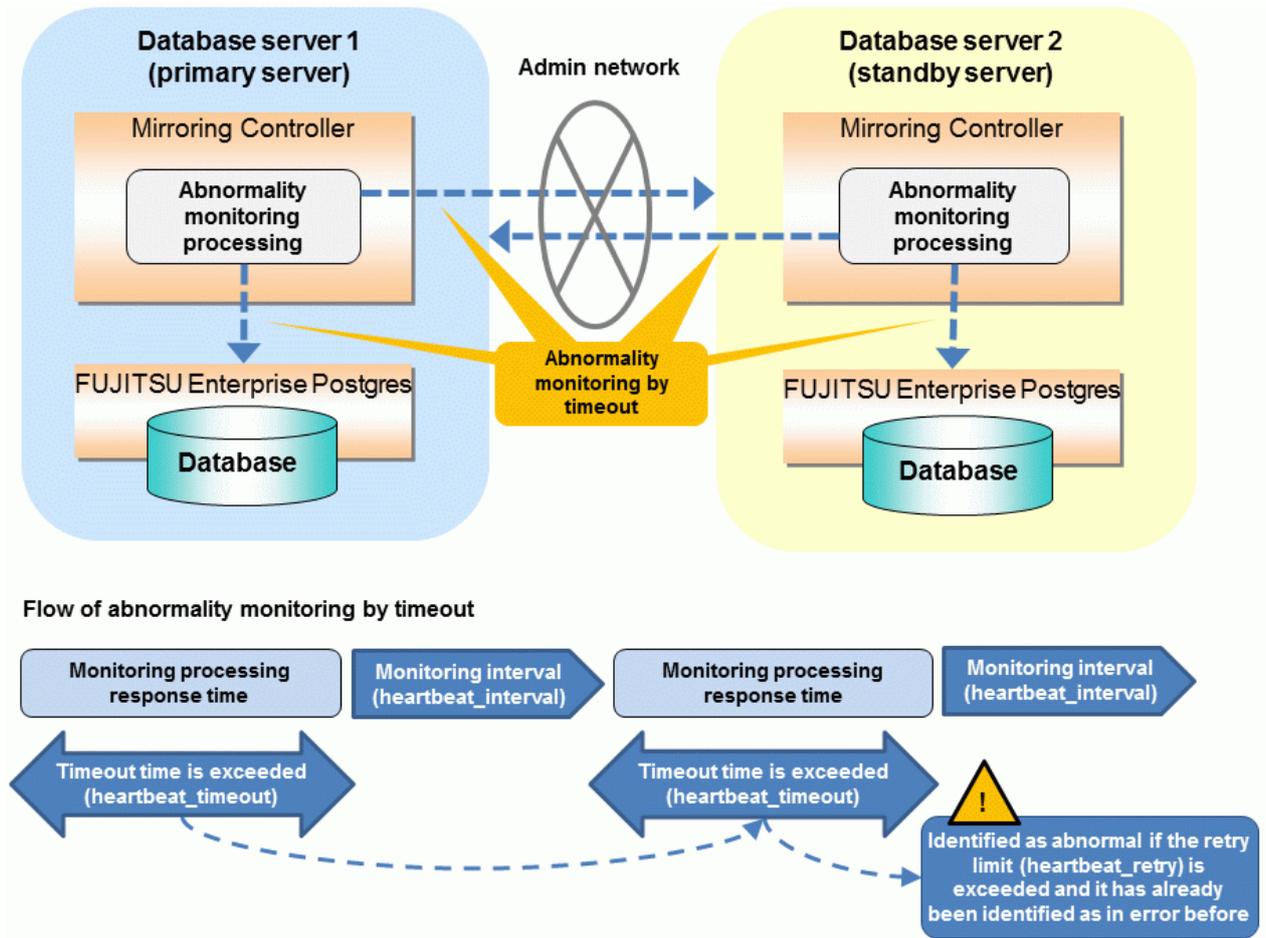
Mirroring Controller uses a monitoring method that outputs an error if the timeout time is exceeded when accessing resources targeted for monitoring. If the timeout time is short, switch/disconnection of the standby server can be performed faster, however, there is greater risk of misdetection, so an appropriate design is required.

You can optimize degrading operation by editing the values for the following parameters in the server definition file in accordance with the system. Refer to "[Appendix A Parameters](#)" for information on how to edit these parameters.

Table 3.6 Parameters

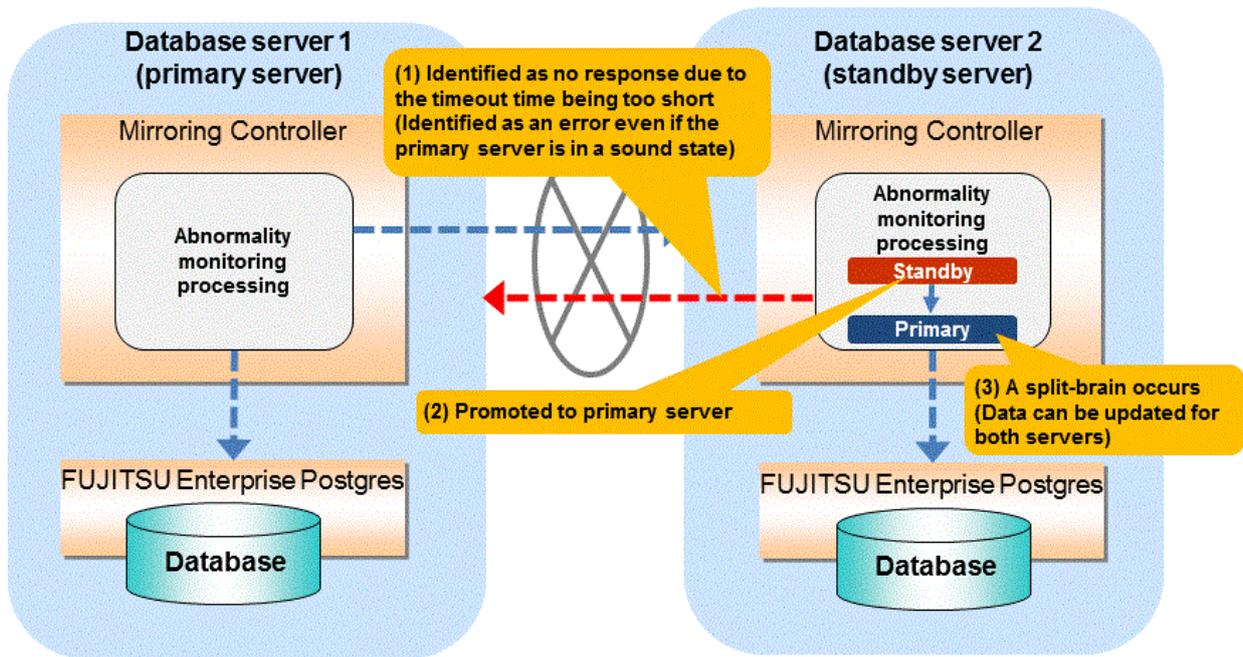
Parameter	Description
Abnormality monitoring interval (<code>heartbeat_interval</code>)	Mirroring Controller is configured so that abnormality monitoring does not place a load on the system. This parameter does not normally need to be set. (The default is 800 milliseconds.)

Parameter	Description
Abnormality monitoring timeout time (heartbeat_timeout)	Set the time during which a load is placed continuously on the server or 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.
Abnormality monitoring retry times (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.)

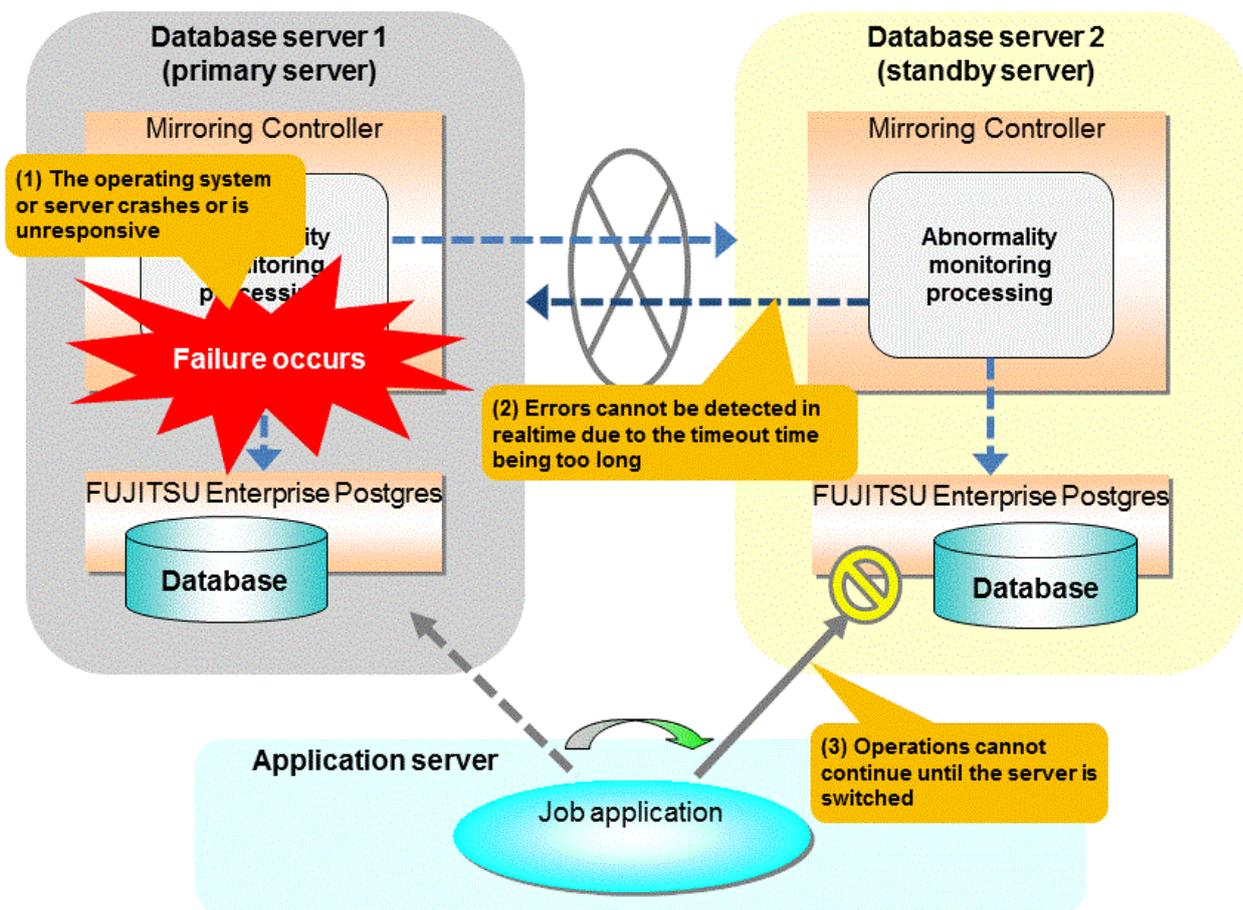


The following type of issue occurs if the tuning related to abnormality monitoring is not performed appropriately.

● If the timeout time is too short



● If the timeout time is too long



Notes regarding monitoring when the operating system or server crashes or is unresponsive

Monitoring is performed upon the aforementioned timeout when 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 "[3.4 Setting up the Standby Server](#)", build (set up) the old primary server as the new standby server, from the new primary server.

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.

3.9 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.

If using an rc script

Perform the following procedure:

1. Set the OS user account of the instance administrator user

If you logged in using the OS user account of the instance administrator user, set the environment variables required for starting the instance.

This setting is required for executing the "su -" command.

2. Edit the shell script

Copy the sample shell script, and edit the environment variables in the shell script to suit the environment.

Path of the sample shell script

```
/installDir/share/rc_mcoi.sample
```

3. Copy the revised shell script and set access permissions

As the OS superuser, copy the revised shell script to /etc/rc.d/init.d, and then set and register the access rights.

In the following example, the shell script was created as "rc_mc_inst1".

Example)

```
# cp rc_mc_inst1 /etc/rc.d/init.d/
# chmod 755 /etc/rc.d/init.d/rc_mc_inst1
```

4. Register and enable automatic start and stop

As the OS superuser, execute the chkconfig command to register and enable the script. To register and enable the script, specify the options shown in the table below.

Option	Description
--add	Registers the shell script as the startup script.
--level	Sets and enables the startup script for each run level.

In the following example, the shell script is registered, and then enabled for run levels 3 and 5.

Example)

```
# chkconfig --add rc_mc_inst1
# chkconfig --level 35 rc_mc_inst1 on
```

If using systemd

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.

Path of the sample file

```
/installDir/share/mcoi.service.sample
```

Example)

In the following example, the installation directory is "/opt/fsepv<XY>server64", and the instance name is "inst1".

```
# cp /opt/fsepv<XY>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	Specifies the feature overview.
		MirroringController <u>instanceName</u>	Specifies the name of the target instance. (*1)
Service	ExecStart	/bin/bash -c ' <u>installDir</u> /bin/mc_std start <u>installDir</u>	Command to be executed when the service is started.

Section	Option	Specified value	Description
		<i>MirroringControllerManagementDir</i> <i>r' mc_ctlOption</i>	Specify the option you want to add when the mc_ctl option is executed 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.
	ExecStop	/bin/bash -c ' <i>installDir</i> /bin/mc_std stop <i>installDir</i> <i>MirroringControllerManagementDir</i> <i>r' mc_ctlOption</i>	Command to be executed when the service is stopped. Specify the option you want to add when the mc_ctl command is executed 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.
	User	<u>User</u>	OS user account of the instance administrator user.
	Group	<u>Group</u>	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. 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
```

3.10 Backup Operation

This section explains the backup operation for database multiplexing mode.

3.10.1 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.

3.10.2 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 5 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 4 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.

4.1 Starting and Stopping Instances

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.

Enabling automatic switch/disconnection

Execute the `mc_ctl` command with the `-f` option specified.

If you start Mirroring Controller and the instance without specifying the `-f` option, start Mirroring Controller and then execute the `mc_ctl` command in enable-failover mode, or restart Mirroring Controller with the `-f` option specified.

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

When only the instance is started and stopped, the following will happen:

- When only the instance is started

Errors indicated in "2.1 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.

4.2 Checking the Database Multiplexing Mode Status

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.

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 /mcdire/inst1
mirroring status
```

```
-----
switchable
server_id host_role      host          host_status  db_proc_status  disk_status
-----
----
nd1       primary      192.0.2.100  normal      normal          normal
nd2       standby     192.0.2.110  normal      normal          normal
```

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_location`, `replay_location`) match, and that they are fully synchronized.

Example)

```
postgres=# select * from pg_stat_replication;
-[ RECORD 1 ]-----+-----
pid          | 14336
usesysid    | 10
username     | fsep
application_name | standby
client_addr  | 192.0.2.210
client_hostname |
client_port  | 27500
backend_start | 2015-04-23 20:24:13.761424+09
backend_xmin |
state        | streaming
sent_location | 0/3000060
write_location | 0/3000060
flush_location | 0/3000060
replay_location | 0/3000060
sync_priority | 1
sync_state   | sync
```



- 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`.

4.3 Manually Switching the Primary Server

The primary server cannot be switched automatically in the following case:

- If automatic switch/disconnection is disabled

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 /mdir/inst1
```

4.4 Manually Disconnecting the Standby Server

The standby server cannot be disconnected automatically in the following case:

- If automatic switch/disconnection is disabled

In this case, to manually disconnect the standby server, execute the mc_ctl command in stop mode on the standby server and comment out the synchronous_standby_names parameter in the postgresql.conf file on the primary server.

1. Execute the mc_ctl 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
```

4.5 Monitoring Mirroring Controller Messages

The messages that are output by Mirroring Controller are output to the system log. 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.



Point

To monitor message types considered to be important, a setting must be configured in the system log 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.

Display format

```
programName[processId]: messageType: messageText (messageNumber)
```

Specify the program name in the syslog_ident parameter of the serverIdentifier.conf file.

The message types output by Mirroring Controller, their severity, and their corresponding value in the system log are shown in the table below.

Table 4.1 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 one of the multiplexed database systems requiring recovery of this system, and also the content and cause of the abnormality.	CRIT

Message type	Severity	Meaning	System log
PANIC		Reports that an abnormality was detected in all multiplexed database systems requiring immediate recovery of this 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.

4.6 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.

4.6.1 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.

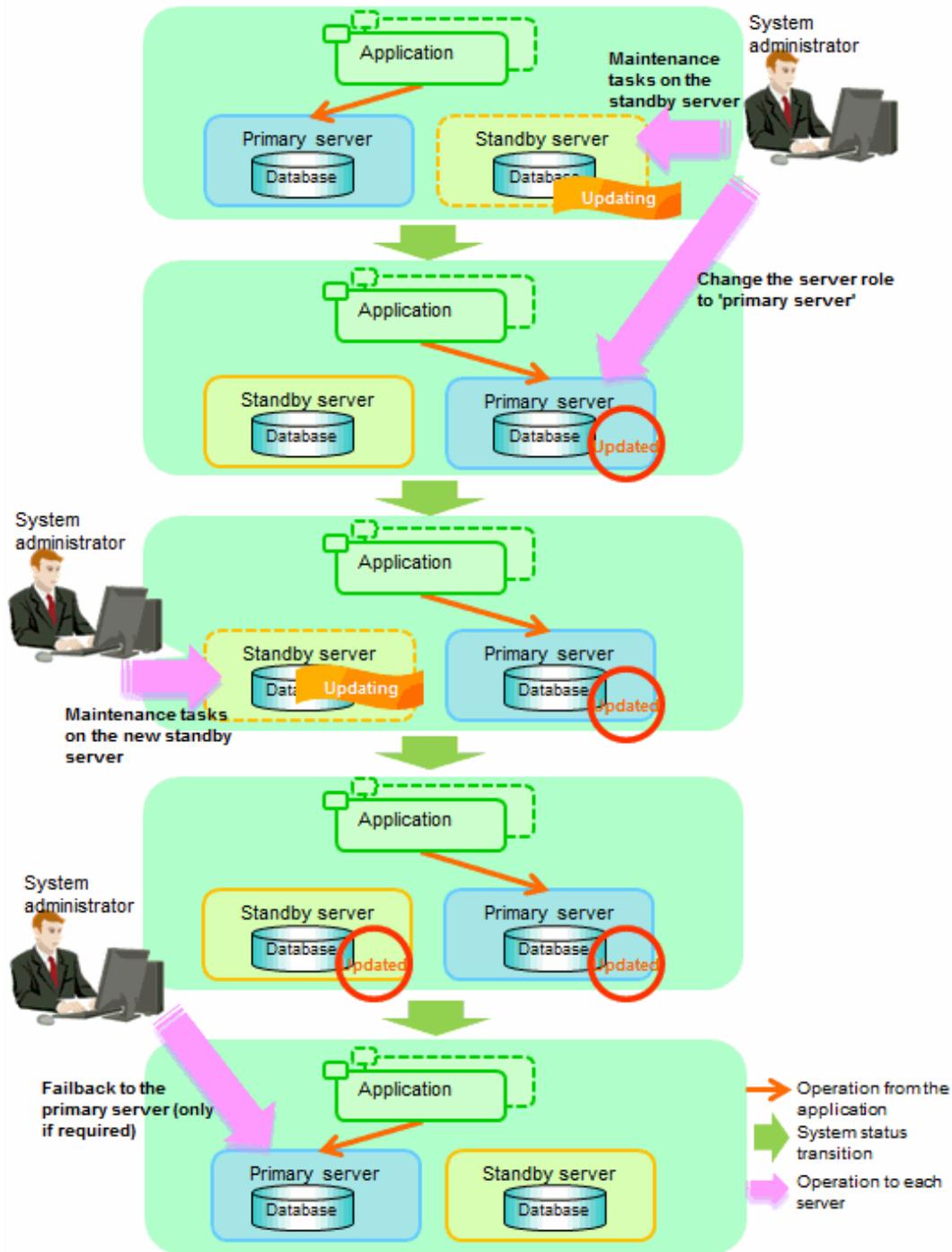


See

.....
 If the downtime due to the maintenance of the standby server is expected to be long, refer to "Standby server downtime" in "4.7.1 Changes Required when the Standby Server is Stopped".

The flow of a rolling update is shown below.

Figure 4.1 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 /mdir/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 "3.9 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.
.....

If using an rc script

As the OS superuser, execute the `chkconfig` command to disable automatic start and stop.

In the example below, the shell script for automatic start and stop was created as "rc_mc_inst1" and is disabled for run levels 3 and 5.

Example)

```
# chkconfig --level 35 rc_mc_inst1 off
```

If using systemd

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 --xlog --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 "3.4.2 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.

If using an rc script

As the OS superuser, execute the `chkconfig` command to enable the shell script.

In the example below, the shell script for automatic start and stop was created as "rc_mc_inst1" and is enabled for run levels 3 and 5.

Example)

```
# chkconfig --level 35 rc_mc_inst1 on
```

If using systemd

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 with the -f option specified. This enables automatic switch and disconnection.

If you start Mirroring Controller and the instance without specifying the -f option, automatic switch and 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 /mcdire/inst1 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode.

Example)

```
$ mc_ctl start -M /mcdire/inst1 -w
```

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 /mcdire/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 /mcdire/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 "3.9 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.

If using an rc script

As the OS superuser, execute the chkconfig command to disable automatic start and stop.

In the example below, the shell script for automatic start and stop was created as "rc_mc_inst1" and is disabled for run levels 3 and 5.

Example)

```
# chkconfig --level 35 rc_mc_inst1 off
```

If using systemd

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 --xlog --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 "3.4.2 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.

If using an rc script

As the OS superuser, execute the chkconfig command to enable the shell script.

In the example below, the shell script for automatic start and stop was created as "rc_mc_inst1" and is enabled for run levels 3 and 5.

Example)

```
# chkconfig --level 35 rc_mc_inst1 on
```

If using systemd

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 with the -w and -f options specified. This enables automatic switch and disconnection.

If you start Mirroring Controller and the instance without specifying the -f option, automatic switch and 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 /mdir/inst1 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode with the -w option specified.

Example)

```
$ mc_ctl start -M /mdir/inst1 -w
```

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 "5.1.1.3 Failback of the Primary Server" for details.



Obtain a backup as soon as this task is complete.

4.6.2 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.

Example)

```
$ mc_ctl stop -M /mdir/inst1 -a
```

4.7 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

4.7.1 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, 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.



.....
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 standby server system log, and all transaction logs that should be transferred to the standby server may be lost.

- The `wal_keep_segments` 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 "[3.8.1 Tuning to Stabilize the Database Multiplexing Mode](#)" for details.



.....
The standby server must be rebuilt in both cases above.

Take the action advised in the recovery operation that starts from "[5.1.1.1.3 Identify cause of error and perform recovery](#)" through to "[5.1.1.2 Rebuild the Standby Server](#)".
.....

4.7.2 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 3 Setting up Database Multiplexing Mode](#)".



If the data storage destination directory name is not comprised of ASCII characters excluding spaces

Stop the application job and then migrate to a directory with a name that uses only ASCII characters excluding spaces:

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 excluding spaces.
-



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 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 excluding spaces in the data storage destination directory.

2. Stop the application jobs

Stop the application jobs to be connected to the primary server.

3. 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 "[3.3.2 Creating, Setting, and Registering the Primary Server Instance](#)" on the primary server.

4. Set up database multiplexing mode on the standby server

Refer to "[3.4.1 Setting up Database Multiplexing Mode on the Standby Server](#)" for details.

5. Create the standby server instance and start it

Refer to "[3.4.2 Creating, Setting, and Registering the Standby Server Instance](#)" for details.

After the above steps are completed, refer to the remaining explanations in "[Chapter 3 Setting up Database Multiplexing Mode](#)" and ensure that the required settings and operations are completed.

4.7.3 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.

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.

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 /mdir/inst1
```



See

Refer to "Security-Related Notes" in the Operation Guide for details on deleting the data securely.

3. 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 /mdir/inst1
```

4. Delete the database multiplexing mode settings that were configured for the primary server instance.

Take the appropriate action for each parameter and resource by editing postgresql.conf as shown below. Also, delete recovery.conf or change its name to recovery.done for example.

File	Parameter	Action
postgresql.conf	wal_level	Delete the "hot_standby" value that was specified.
	max_wal_senders	Revert the value to the one set before database multiplexing mode was set.
	synchronous_standby_names	Delete.
	wal_sender_timeout	
	wal_receiver_timeout	
	listen_addresses	Revert the value to the one set before database multiplexing mode was set.
	max_connections	
	synchronous_standby_names	Delete.
hot_standby		

- Mirroring Controller management directory

Additionally, 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

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 "3.10 Backup Operation" for details on the backup operation.
- Refer to "Appendix A Parameters" for details on the postgresql.conf file parameters.

Chapter 5 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

5.1 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 /mdir/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 /mdir/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 "5.1.3 Addressing Errors During Degrading Operation", and then recover to database multiplexing mode.

5.1.1 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

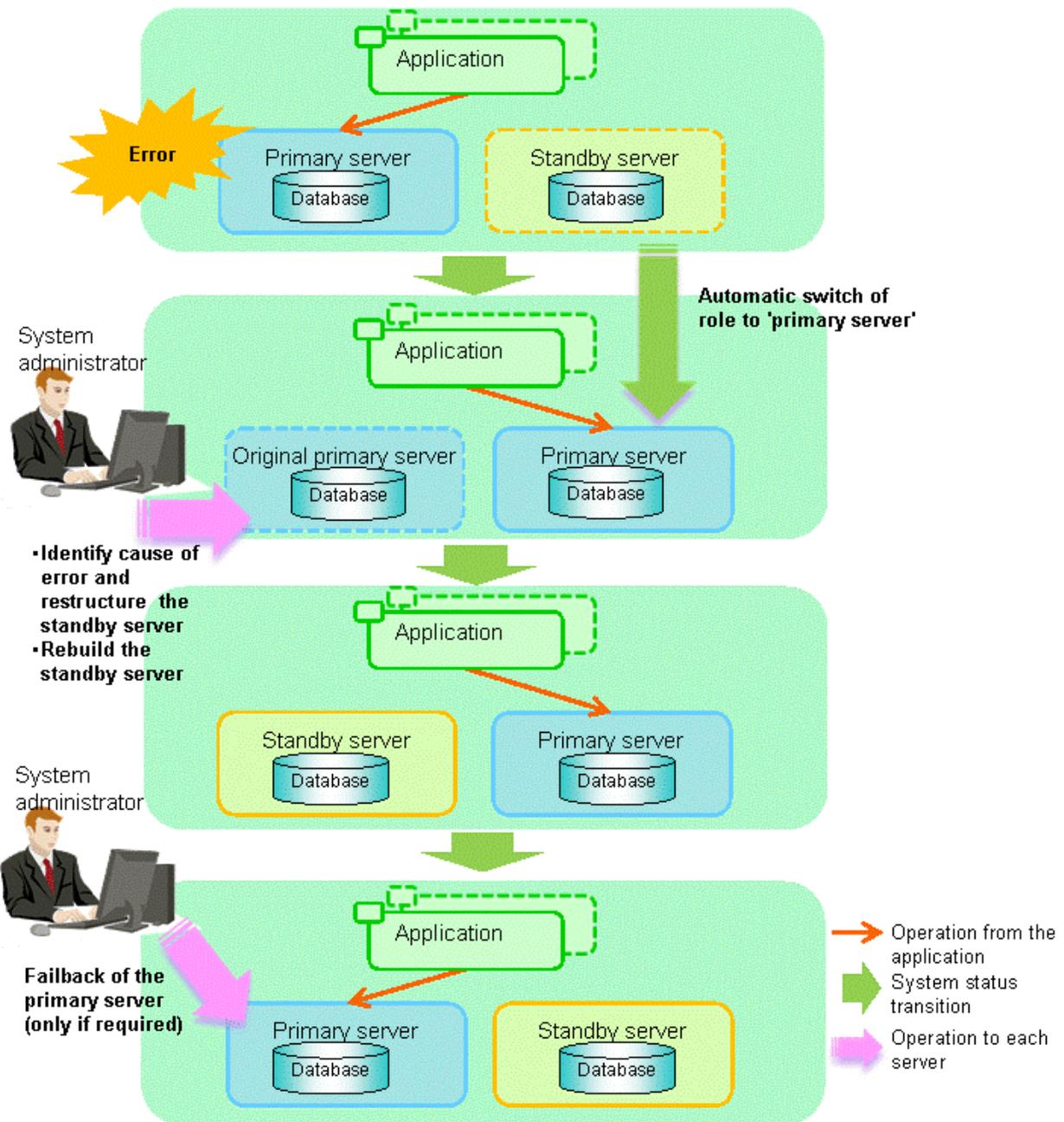
- After a switch has occurred as a result of an abnormality on the primary server, the database will not have a multiplexed configuration until the standby server is rebuilt. Remove the cause of the error as quickly as possible, and then rebuild the standby server.
 - If the reference job was executed 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 5.1 Flow of operations



5.1.1.1 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](#)

5.1.1.1.1 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 /mdir/inst1
```

This also stops the instance that is required to perform the recovery.



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.

5.1.1.1.2 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.

5.1.1.1.3 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:

- `pg_basebackup`

Creates a copy of all resources of the primary server instance. This command is used for example when creating the first instance on the standby server.

- `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 true:

- a. The original primary server was stopped normally.
- b. Checksums were enabled when an instance was created.
- c. 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.

If it is required to start the original primary server again to then stop it normally, both servers will temporarily operate as primary servers. If data is updated on the original primary server, data corruption will occur, so take measures such as changing settings in `pg_hba.conf` to disallow client connections.



Refer to "pg_rewind" under "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. Use the `pg_ctl` command to start the original primary server instance.

Example)

```
$ pg_ctl start -w -D /database/inst1
```

2. Use the `pg_ctl` command to stop the original primary server instance.

Example)

```
$ pg_ctl stop -D /database/inst1
```

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 --source-server='user=userName host=newPrimaryServerHostName  
port=newPrimaryServerPortNumber'
```

5.1.1.2 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 with the `-f` option specified. This 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 /mdir/inst1 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

```
$ mc_ctl start -M /mdir/inst1 -w
```

5.1.1.3 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 /mdir/inst1
```

After executing the `mc_ctl` command in switch mode, the status will be as follows:

Example)

```
$ mc_ctl status -M /mdir/inst1  
mirroring status  
-----  
switched  
server_id host_role host host_status db_proc_status disk_status  
-----  
nd1 primary 192.0.2.100 normal abnormal(postmaster) normal  
nd2 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 /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 --xlog --progress --verbose -R --  
dbname='application_name=standbyServerName' -h primaryServerHostName -p primaryServerPortNumber
```



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 "3.4.2 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 with the `-w` and `-f` options specified. This 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 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the `mc_ctl` command in start mode with the `-w` option specified.

Example)

```
$ mc_ctl start -M /mcdir/inst1 -w
```

5.1.2 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.



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.

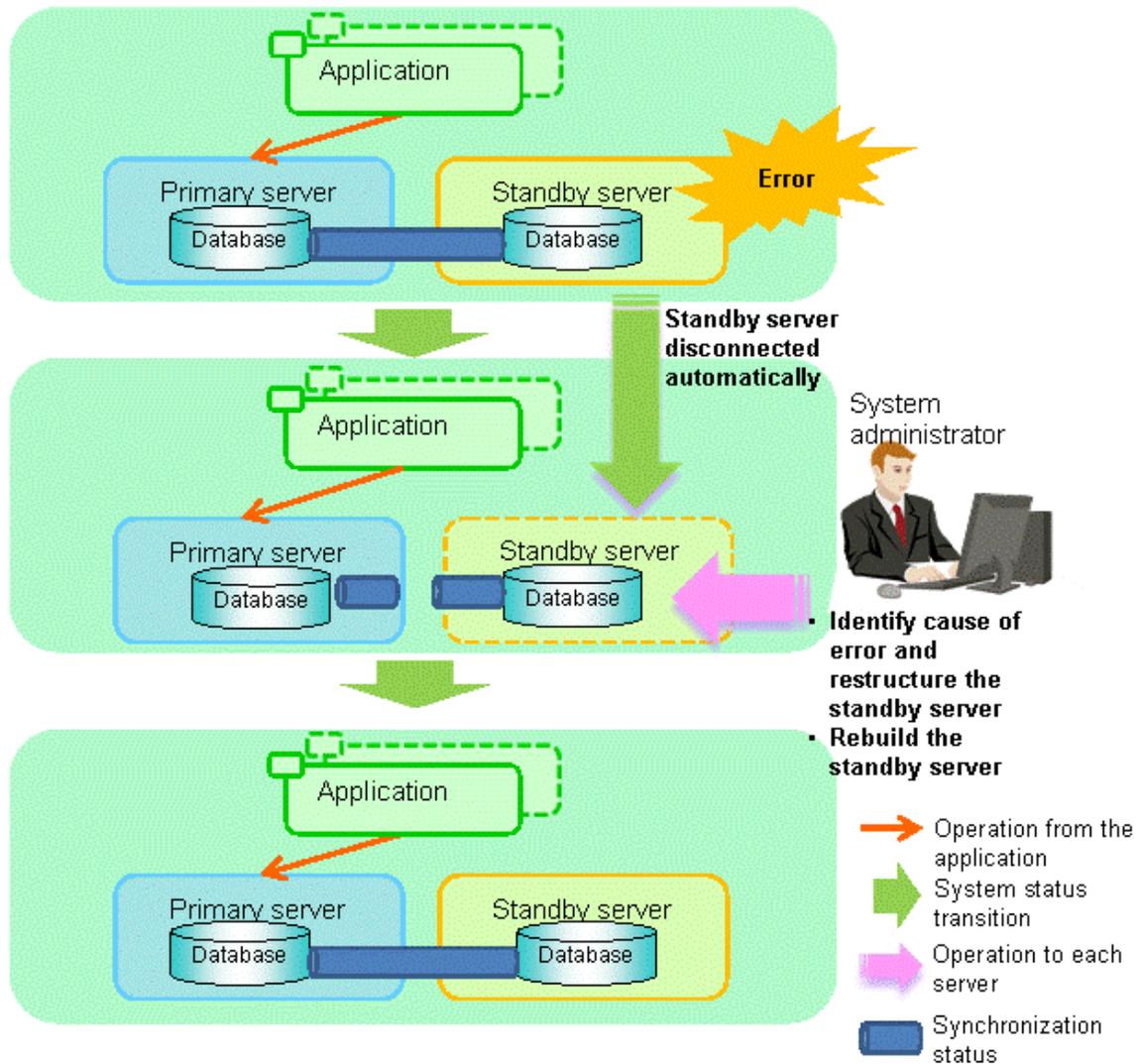
If the disconnection occurred and the server has started degrading, perform the following operations to recover the standby server and revert it to its original state:

- [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 5.2 Flow of operations



5.1.2.1 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](#)

5.1.2.1.1 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 /mdir/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.

5.1.2.1.2 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.

5.1.2.1.3 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 --xlog --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 "3.4.2 Creating, Setting, and Registering the Standby Server Instance", and then perform the recovery.

5.1.2.2 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 with the `-f` option specified. This 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 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

```
$ mc_ctl start -M /mcdir/inst1 -w
```

5.1.3 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 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](#)

5.2 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:

- Execute the mc_ctl command in switch mode

Execute the mc_ctl command in switch mode on the primary server or the standby server.

Example)

```
$ mc_ctl switch -M /mcdire/inst1
```



Note

If the server fails during execution of the mc_ctl command, automatic switch, or automatic disconnection, or if communication between the primary server and the standby server is lost, a message indicating that the mc_ctl command is being executed may be output even though the command is not running elsewhere, and the command may end abnormally.

In this case, switch the server in accordance with the "Execute the mc_ctl command in stop mode" procedure below.

- Execute the mc_ctl command in stop mode

Execute the mc_ctl command in stop mode on the server on which the error occurred.

Example)

```
$ mc_ctl stop -M /mcdire/inst1
```

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 mc_ctl command to forcibly stop Mirroring Controller.

Example)

```
$ mc_ctl stop -M /mcdire/inst1 -e
```



See

Recovery to database multiplexing mode

Refer to "[5.1.1.2 Rebuild the Standby Server](#)" and "[5.1.1.3 Failback of the Primary Server](#)" for information on recovery to database multiplexing mode.

5.3 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.

In this case, perform the following procedure to disconnect the primary server manually.

1. On the standby server, execute the `mc_ctl` command in stop mode with the `-e` option specified.

Example)

```
$ mc_ctl stop -M /mdir/inst1 -e
```

2. On the primary server, comment out the `synchronous_standby_names` parameter in the `postgresql.conf` file.

Example)

```
:  
# synchronous_standby_names = 'standby1 '
```

3. Execute the `pg_ctl` command in reload mode on the primary server.

Example)

```
$ pg_ctl reload -D /database/inst1
```

 See

.....
Refer to "pg_ctl" in Reference for information on the command.
.....

 See

Recovery to database multiplexing mode

Refer to "5.1.2.2 Rebuild the Standby Server" for information on recovery to database multiplexing mode.
.....

5.4 Action Required when All Servers or Instances Stopped

This section explains what happens when all servers or instances on the server have stopped, so jobs cannot continue.

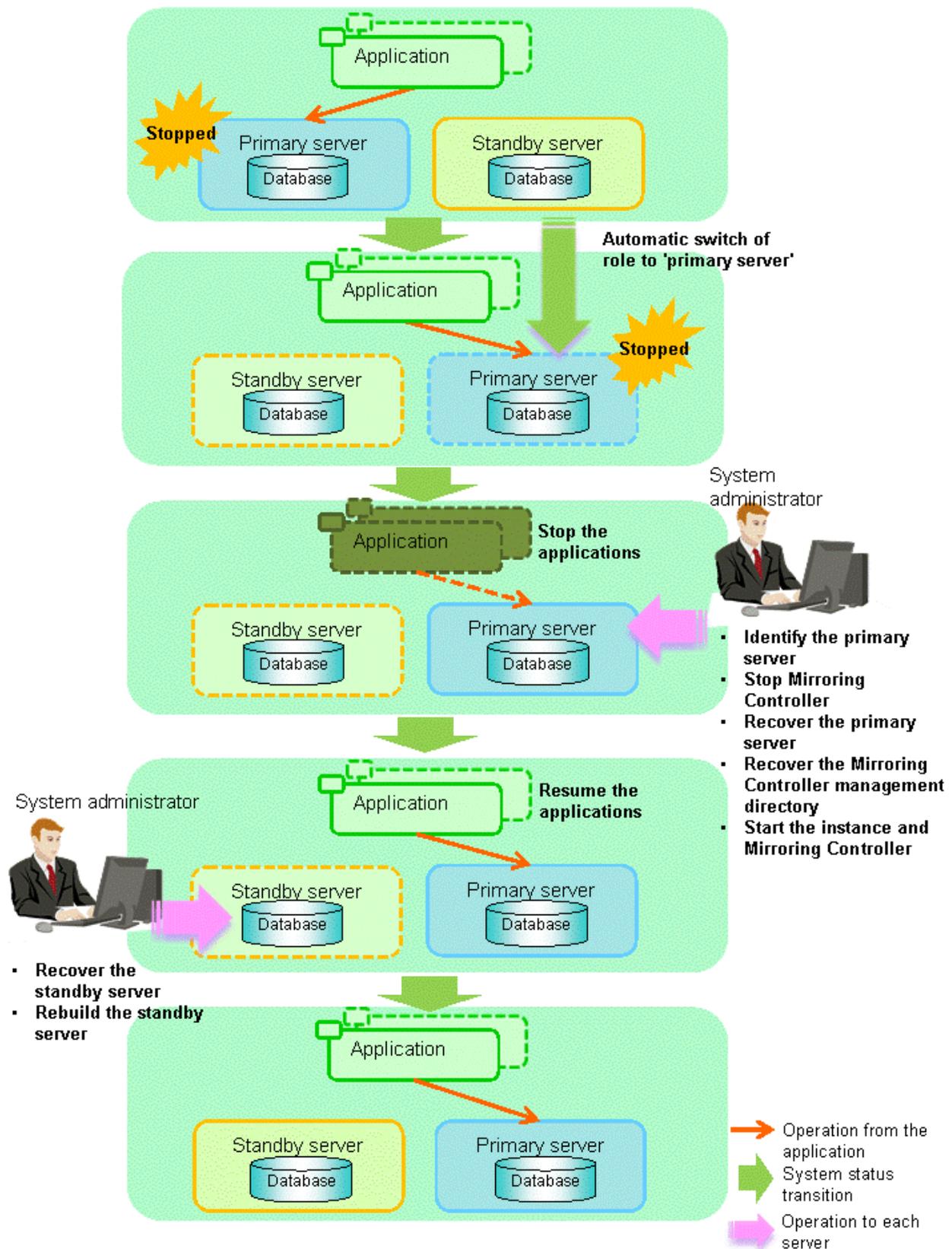
 See

Recovery to database multiplexing mode

Refer to "5.1.1.2 Rebuild the Standby Server" and "5.1.1.3 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 5.3 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.

Example)

```
$ mc_ctl stop -M /mdir/inst1
```



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 /mdir/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 "[3.10.2 Database Backup Operation](#)", perform the following procedure for the recovery:

- 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/*_xlog
```

- 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.
- If the latest archive log exists on the standby server, copy (*1) the archive log and overwrite (*2) it to the backup storage destination directory on the primary server.
- Execute the `pgx_rcvall` command on the primary server, specifying the backup storage destination directory you used in step 3.



*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.



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 with the `-f` option specified. This 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 /mdir/inst1 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

```
$ mc_ctl start -M /mdir/inst1 -w
```

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.

Example)

```
$ mc_ctl stop -M /mdir/inst1
```

9. Recover the standby server

Refer to "[3.4.2 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 with the `-f` option specified. This 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 /mdir/inst1 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

```
$ mc_ctl start -M /mcdir/inst1 -w
```

5.5 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 with the `-f` option specified. This 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 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

```
$ mc_ctl start -M /mcdir/inst1 -w
```

6. Build the new standby server

Refer to "[3.4 Setting up the Standby Server](#)" for information on building (setting up) a standby server from the primary server.

Chapter 6 Managing Mirroring Controller Using WebAdmin

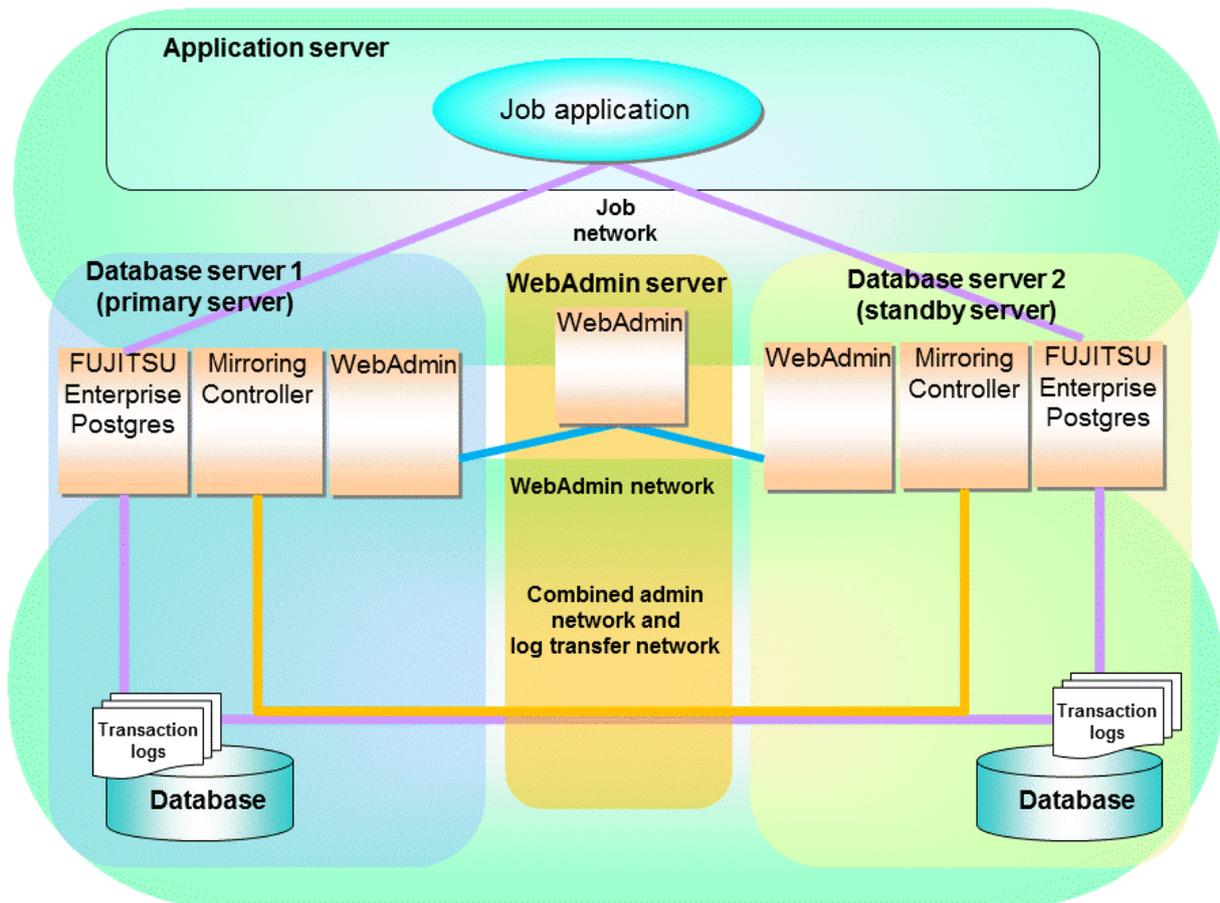
This chapter describes how to set up and manage Mirroring Controller in a streaming replication cluster using WebAdmin.

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 having master and synchronous standby instances.

The configuration of the database multiplexing system built using WebAdmin is shown below:

Figure 6.1 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].

6.1 Mirroring Controller Setup

Perform the following procedure to setup Mirroring Controller in a streaming replication cluster. The option for this setup is accessible only in synchronous standby instance.

1. In the [Instances] tab, select the synchronous standby instance on which Mirroring Controller needs to be set up.
2. Click .

3. Enter the information for the Mirroring Controller to be setup.

In the example below, Mirroring Controller is being setup for the replication cluster having master instance "inst1" and standby instance "inst1s".

The screenshot displays the 'Mirroring Controller setup' page in the Fujitsu Enterprise Postgres web interface. The page is divided into two main sections: 'Mirroring Controller setup for master instance' and 'Mirroring Controller setup for standby instance'. At the top, there is a toggle for 'Enable automatic switch over' set to 'Yes'. The master instance configuration includes fields for Instance name (inst1), Instance port (27500), Instance host (localhost), Mirroring Controller port (27505), Mirroring Controller management directory (/home/fsep/test/mkdir/inst1), Heartbeat interval (800), Heartbeat timeout (1), and Heartbeat retry (2). The standby instance configuration includes fields for Instance name (inst1s), Instance port (27502), Instance host (fepwa-ae1), Mirroring Controller port (27505), Mirroring Controller management directory (/home/fsep/test/mkdir/inst1s), Heartbeat interval (800), Heartbeat timeout (1), and Heartbeat retry (2).

Mirroring Controller setup for master instance	
Instance name	inst1
Instance port	27500
Instance host	localhost
Mirroring Controller port *	27505
Mirroring Controller management directory *	/home/fsep/test/mkdir/inst1
Heartbeat interval (in milliseconds) *	800
Heartbeat timeout (in seconds) *	1
Heartbeat retry *	2

Mirroring Controller setup for standby instance	
Instance name	inst1s
Instance port	27502
Instance host	fepwa-ae1
Mirroring Controller port *	27505
Mirroring Controller management directory *	/home/fsep/test/mkdir/inst1s
Heartbeat interval (in milliseconds) *	800
Heartbeat timeout (in seconds) *	1
Heartbeat retry *	2

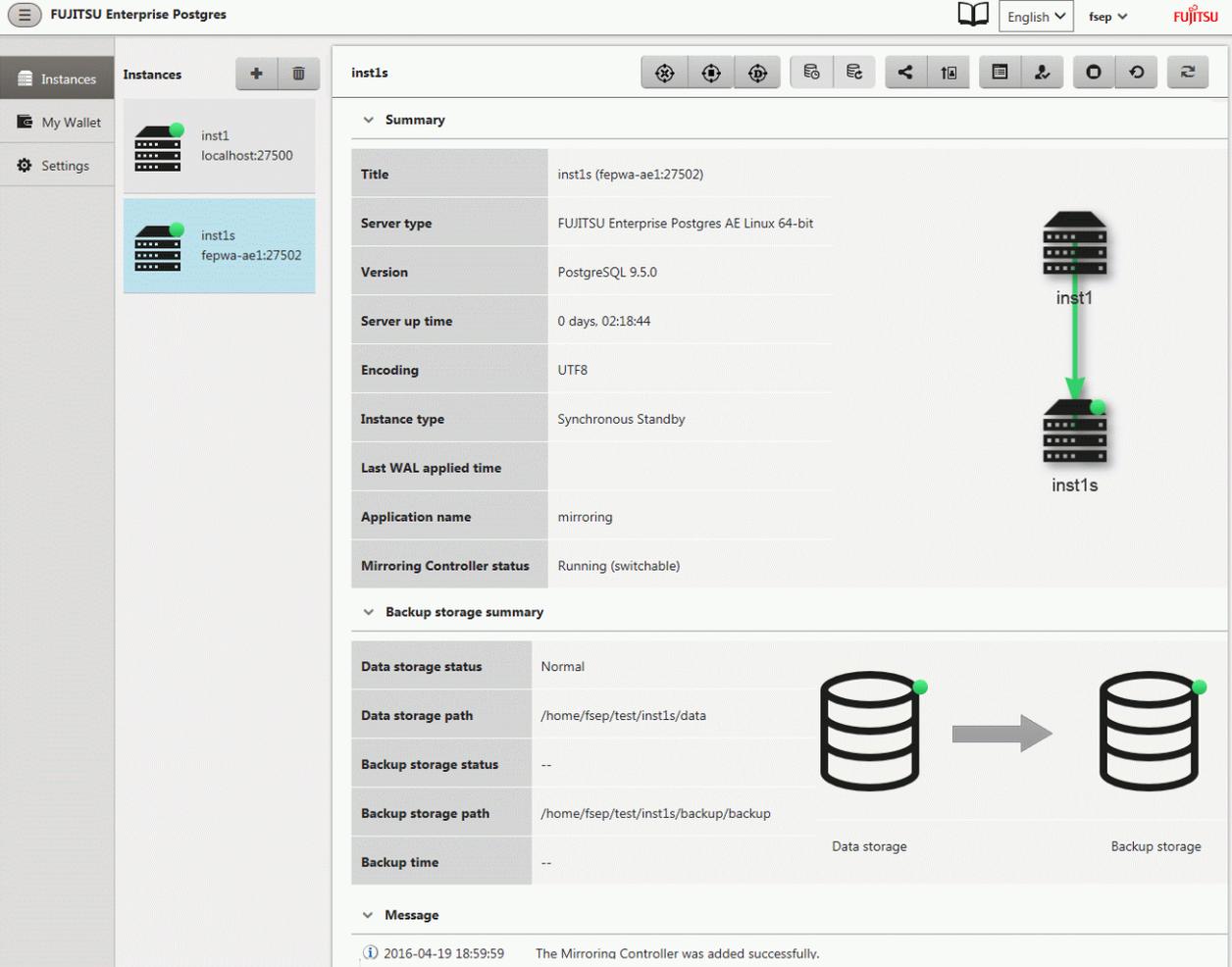
The instance name, host address and port of the master and standby instances are displayed for easy reference.

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 port]: Port number of Mirroring Controller.
- [Mirroring Controller management directory]: Directory where the Mirroring Controller configuration files will be stored.
- [Heartbeat interval (in milliseconds)]: Number of milliseconds between two consecutive heartbeat checks. The default is "800".
- [Heartbeat timeout (in 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".

4. Click  to setup Mirroring Controller.
5. Upon successful completion, Mirroring Controller will be started on master and standby instances.

In the [Instances] tab, select "inst1s". The following page will be displayed, with the Mirroring Controller status:



The screenshot displays the Fujitsu Enterprise Postgres WebAdmin interface. The left sidebar shows the 'Instances' tab with two instances: 'inst1' (localhost:27500) and 'inst1s' (fepwa-ae1:27502). The main content area shows the configuration for 'inst1s'.

Summary	
Title	inst1s (fepwa-ae1:27502)
Server type	FUJITSU Enterprise Postgres AE Linux 64-bit
Version	PostgreSQL 9.5.0
Server up time	0 days, 02:18:44
Encoding	UTF8
Instance type	Synchronous Standby
Last WAL applied time	
Application name	mirroring
Mirroring Controller status	Running (switchable)

Below the summary, there is a diagram showing a green arrow pointing from 'inst1' to 'inst1s', indicating the mirroring relationship.

Backup storage summary	
Data storage status	Normal
Data storage path	/home/fsep/test/inst1s/data
Backup storage status	--
Backup storage path	/home/fsep/test/inst1s/backup/backup
Backup time	--

The backup storage summary includes a diagram showing two database icons: 'Data storage' and 'Backup storage', with an arrow pointing from 'Data storage' to 'Backup storage'.

Message

2016-04-19 18:59:59 The Mirroring Controller was added successfully.

6.2 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.

6.3 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.



4. In the confirmation dialog box, 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.

6.4 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.

6.5 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.

6.6 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].

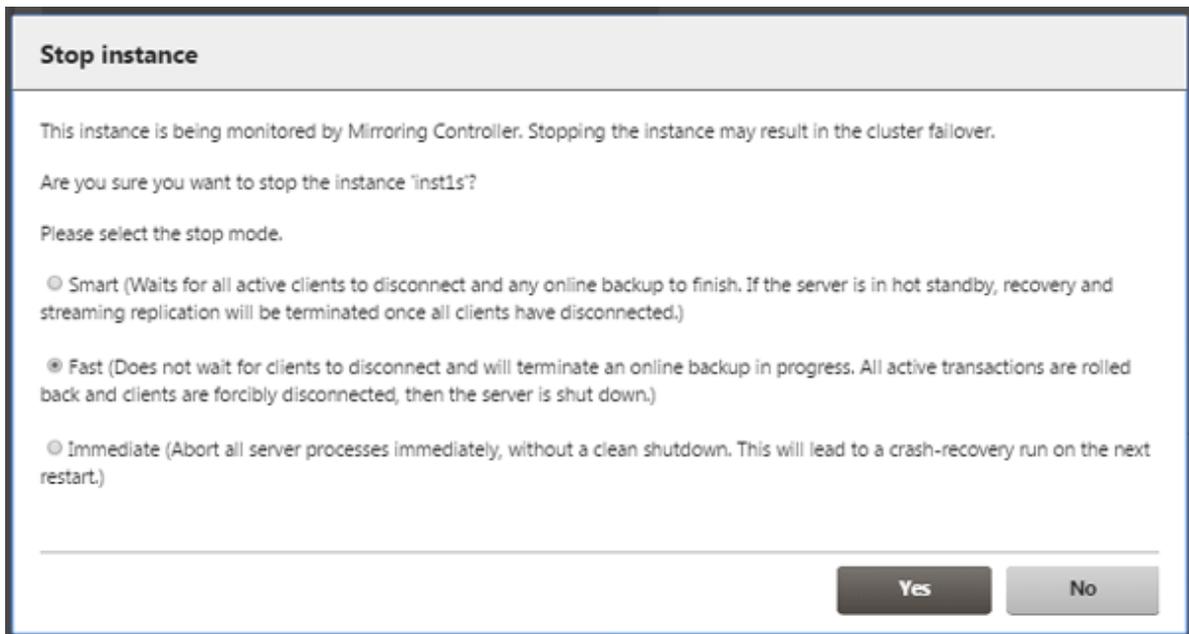
Mirroring Controller setup will be removed from the cluster. The cluster status will be updated and a confirmation message entry will be displayed in the [Message] section.

6.7 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 the 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.

6.8 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 refers to a situation where data update is performed on both servers because they both work as a primary server temporarily.

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 "[3.8.4 Tuning for Optimization of Degrading Operation Using Abnormality Monitoring](#)" 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 primary instance is selected in the [Instances] tab, and
4. "Standalone" is displayed in "Standalone".



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.

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	Literal "hot_standby"	Specify the output level for the transaction log. Specify "hot_standby".
max_wal_senders	Number of standby servers + 1	Specify the number of standby servers + 1. In FUJITSU Enterprise Postgres, 1:1 standby is used, so specify 2.
synchronous_standby_names	' <i>standbyServerName</i> '	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.
hot_standby	on	Specify whether queries can be run on the standby server. Specify this to execute reference jobs on the standby server. (*1)
wal_keep_segments	Number of file segments	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. Setting this parameter requires consideration regarding stabilization of the database multiplexing mode (refer to " 3.8.1 Tuning to Stabilize the Database Multiplexing Mode " for details). Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on estimating the WAL segment.
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 of (heartbeat_interval + heartbeat_timeout/1000) x (heartbeat_retry + 1) of the <i>serverIdentifier.conf</i> file, you can unify the time after which it is determined that an error has occurred.

Parameter	Value set	Explanation
wal_receiver_timeout	Timeout (milliseconds)	Specify the time period after which it is determined that an error has occurred in the transaction log transfer on the standby server. By aligning this value with the value of (heartbeat_interval + heartbeat_timeout) x heartbeat_retry of the <i>serverIdentifier.conf</i> file, 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	<i>'installDir/bin/pgx_xlogcopy.cmd "%p" "backupDataStorageDestinationDirectory/archived_xlog/%f"'</i>	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	The limit value for the number of standby server connections + number of simultaneous executions of mc_ctl status (*2) + 2 + 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.

*1: Mandatory to use the Mirroring Controller.

*2: 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
synchronous_standby_names	<i>'primaryServerName'</i>	Use single quotation marks (') to enclose the name that will identify the primary server. Any name can be specified.

Parameter	Value set	Explanation
		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.
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_command	'installDir/bin/pgx_xlogcopy.cmd "%p" " backupDataStorageDestinationDirectory/archived_xlog/%f"'	Specify the command and storage destination to save the transaction log.

A.3 Network Configuration File

Define the network configuration that will link the Mirroring Controller processes in the network configuration file (network.conf file).

Define the same content on the primary server and standby server.

Table A.3 network.conf file

Format specified	Explanation
<i>serverIdentifier hostName portNumber</i> <i>Or</i> <i>serverIdentifier ipAddress portNumber</i>	<p>Specify any identifier (*1) for the server, a host name (or IP address), and port number (*2), using a space as the separator.</p> <p>Specify the IP address or corresponding host name (*3) that will be used to connect to the admin network.</p> <p>Define the same content on the primary server and standby server.</p> <p>Example)</p> <p>The literal <i><space></i> represents a space.</p> <p>[For IPv4]</p> <pre>server1 <space>192.0.2.100<space>27540 server2 <space>192.0.2.110<space>27540</pre> <p>[For IPv6]</p> <pre>server1 2001:258:8404:1217:250:56ff:fea7:559f 27000 server2 2001:258:8404:1217:250:56ff:fea7:55a0 27100</pre>

*1: The maximum length of the server identifier is 64 bytes. Use ASCII characters excluding spaces to specify this parameter.

*2: 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. Note that the value specified in this parameter must also be set in the /etc/services file.

*3: Use ASCII characters excluding spaces to specify the host name.

A.4 Server Configuration File

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	<i>'dataStorageDestinationDir'</i> [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 excluding spaces to specify this parameter.
db_instance_password	<i>'passwordOfInstanceAdminUser'</i>	Specify the password used when Mirroring Controller connects to a database instance. Use ASCII characters excluding spaces 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. Enclose the name within single quotation marks (').
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. The value 0 indicates that there is no timeout limit. The default is 70000 milliseconds (70 seconds).
core_file_path	<i>CoreFileOutputDir</i>	Specify the directory to which the core file is to be output. Use ASCII characters excluding spaces 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)	<i>'programName'</i>	Specify using single quotation marks (') to enclose the program name used to identify the Mirroring Controller message in the system log. Use ASCII characters excluding spaces to specify this parameter. The default is 'MirroringControllerOpen'.
heartbeat_interval	Interval time for operating system/ server and process heartbeat monitoring, and disk abnormality monitoring (milliseconds)	Servers will contact each other and perform abnormality monitoring at the specified interval. Specify a value between 1 and 2147483647.

Parameter	Value set	Explanation
		The default is 800 milliseconds.
heartbeat_timeout	Timeout for operating system/server and process heartbeat monitoring, and disk abnormality monitoring (seconds)	<p>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.</p> <p>Specify a value between 1 and 2147483647.</p> <p>The default is 1 second.</p>
heartbeat_retry	Number of retries for operating system/server and process heartbeat monitoring, and disk abnormality monitoring (number of times)	<p>Specify the number of retries when an error has been detected that requires the primary server to be switched, or the standby server to be disconnected.</p> <p>If an error is detected in succession more than the specified number of times, switch or disconnection will be performed.</p> <p>Specify a value between 0 and 2147483647.</p> <p>The default is 2 times.</p>

*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.

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.

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 swith/disconnection

Start Mirroring Controller of the server "server1":

Example)

```
$ mc_ctl start -M /mcdire/inst1 -w -f --local-server server1
```

Stop Mirroring Controller of the server "server1":

Example)

```
$ mc_ctl stop -M /mcdire/inst1 --local-server server1
```

Disabling automatic swith/disconnection

Start Mirroring Controller of the server "server1":

Example)

```
$ mc_ctl start -M /mcdire/inst1 -w --local-server server1
```

Stop Mirroring Controller of the server "server1":

Example)

```
$ mc_ctl stop -M /mcdire/inst1 --local-server server1
```

 Note

If using an rc script

Append the `--local-server` option to the `mc_ctl` command call defined in the automatic start shell script.

If using `systemd`

Add the `--local-server` option to the `mc_ctl` option specification for `ExecStart` and `ExecStop` of the unit file for `systemd`.

Refer to "[3.9 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances](#)" for details.

Index

[A]	[M]
About Database Multiplexing Mode..... 4	Manually Disconnecting the Standby Server..... 31
Action Required when All Servers or Instances Stopped.....51	Manually Switching the Primary Server..... 31
Action Required when an Error Occurs in the Database Multiplexing Mode.....42	Matching the system times.....12
Action Required when Automatic Disconnection Fails..... 51	Mirroring Controller Resources.....7
Action Required when Automatic Switch Fails..... 50	Monitoring Mirroring Controller Messages.....32
Action Required when Server Degradation Occurs..... 42	Monitoring using database multiplexing mode..... 4
Addressing Errors During Degrading Operation.....49	
Application Connection Server Settings.....22	[N]
Authentication of the Standby Server.....10	network.conf file.....65
	Network Configuration File.....65
[B]	Notes on CPU Architecture and Products..... 9
Backing up Database Multiplexing Mode Information.....28	Notes on Executing Reference Jobs on the Standby Server.....9
Backup Data Storage Destination Directory..... 68	Notes on Indexes..... 22
Backup Operation..... 28	
	[O]
[C]	Operations in Database Multiplexing Mode.....30
Changes in Operation 38	Operations when the Server has Started Degrading after a Disconnection has Occurred.....47
Changes Required when the Standby Server is Stopped.....39	Operations when the Server has Started Degrading after a Switch has Occurred.....42
Changing from Database Multiplexing Mode to Single Server Mode.....40	Overview of Database Multiplexing Mode..... 4
Changing from Single Server Mode to Database Multiplexing Mode.....39	
Checking the Behavior..... 22	[P]
Checking the Database Multiplexing Mode Status..... 30	Parameters..... 63
Cluster Operation Provided by FUJITSU Enterprise Postgres...2	Parameters Set on the Primary Server..... 63
Configuration for Database Multiplexing Mode..... 6	Parameters Set on the Standby Server.....64
Confirming the Streaming Replication Status.....21	postgresql.conf file..... 63,64
Creating, Setting, and Registering the Primary Server Instance15	Preparing the Backup Disk..... 13
Creating, Setting, and Registering the Standby Server Instance19	
Creating Applications..... 22	[R]
	Rebuild the Standby Server..... 46,49
[D]	Recovering from an Incorrect User Operation..... 55
Database Backup Operation..... 28	Recovery of the Mirroring Controller management directory 45,49
Definition of Cluster Operation..... 2	Redundancy of the Admin and Log Transfer Networks.....8
	Rolling Updates..... 33
[E]	
Encryption of Transaction Logs Transferred to the Standby Server 11	[S]
	Scalability..... 2
[F]	Security in Database Multiplexing..... 9
Failback of the Primary Server..... 46	ServerConfiguration File..... 66
	serverIdentifier.conf file..... 66
[H]	Server Maintenance..... 33
High availability..... 2	Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances.....26
How to Execute the mc_ctl Command..... 68	Setting up Database Multiplexing Mode..... 12
	Setting up Database Multiplexing Mode on the Primary Server13
[I]	Setting up Database Multiplexing Mode on the Standby Server18
Identify cause of error and perform recovery.....45,49	Setting up the Primary Server..... 13
Identify Cause of Error and Restore the Standby Server.....44	Setting up the Standby Server..... 18
Identify Cause of Error and Restructure the Standby Server... 48	Setup..... 12
Installation..... 13	Starting and Stopping Instances..... 30
	Stop Mirroring Controller.....44,48
[L]	Stopping for Maintenance 38
Log shipping mode..... 2	Supplementary Information on Building the Primary Server and Standby Server on the Same Server..... 68

[T]

Tuning.....22

Tuning for Optimization of Degrading Operation Using
Abnormality Monitoring.....23

Tuning to Stabilize Queries on the Standby Server.....23

Tuning to Stabilize Queries on the Standby Server (when
Performing Frequent Updates on the Primary Server)..... 23

Tuning to Stabilize the Database Multiplexing Mode.....22

[U]

Users who perform setup and operations..... 12

[W]

What is a cluster system?2

FUJITSU Enterprise Postgres 9.5



Cluster Operation Guide

Windows

B1WS-1280-01ENZ0(00)
June 2016

Preface

Purpose of This Document

This document explains the items required to operate FUJITSU Enterprise Postgres on a cluster system.

Intended Readers

This document is aimed at people who install and operate FUJITSU Enterprise Postgres on a cluster system. 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:

Part 1 Overview of Cluster Systems

Chapter 1 Definition of Cluster Operation

Provides an overview of cluster operations and their types.

Part 2 Database Multiplexing Mode

Chapter 2 Overview of Database Multiplexing Mode

Provides an overview of database multiplexing mode.

Chapter 3 Setting up Database Multiplexing Mode

Describes how to set up database multiplexing mode.

Chapter 4 Operations in Database Multiplexing Mode

Explains periodic database multiplexing mode.

Chapter 5 Action Required when an Error Occurs in Database Multiplexing Mode

Explains the action required when an error occurs during a database multiplexing mode.

Chapter 6 Managing Mirroring Controller Using WebAdmin

Explains how to setup 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 Supplementary Procedure on Configuring for Operation in Database Multiplexing Mode

Explains the configuration procedure for operation in database multiplexing 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

First edition: July 2016

Copyright

Copyright 2016 FUJITSU LIMITED

Contents

Part 1 Overview of Cluster Systems.....	1
Chapter 1 Definition of Cluster Operation.....	2
1.1 What Is a Cluster System?.....	2
1.2 Cluster Operation Provided by FUJITSU Enterprise Postgres.....	2
Part 2 Database Multiplexing Mode.....	3
Chapter 2 Overview of Database Multiplexing Mode.....	4
2.1 What is Database Multiplexing Mode.....	4
2.2 System Configuration for Database Multiplexing Mode.....	6
2.2.1 Mirroring Controller Resources.....	7
2.2.2 Redundancy of the Admin and Log Transfer Networks.....	8
2.2.3 Notes on CPU Architecture and Products.....	9
2.2.4 Notes on Executing Reference Jobs on the Standby Server.....	9
2.3 Security in Database Multiplexing.....	9
2.3.1 Authentication of the Standby Server.....	11
2.3.2 Encryption of Transaction Logs Transferred to the Standby Server.....	12
Chapter 3 Setting up Database Multiplexing Mode.....	13
3.1 Installation.....	14
3.2 Preparing for Setup.....	14
3.2.1 Preparing the Backup Disk.....	14
3.2.2 Preparatory Tasks for the Output of Error Logs to the Event Log.....	14
3.2.3 Security Policy Settings.....	15
3.3 Setting up the Primary Server.....	15
3.3.1 Setting up Database Multiplexing Mode on the Primary Server.....	15
3.3.2 Creating, Setting, and Registering the Primary Server Instance.....	19
3.4 Setting up the Standby Server.....	22
3.4.1 Setting up Database Multiplexing Mode on the Standby Server.....	22
3.4.2 Creating, Setting, and Registering the Standby Server Instance.....	25
3.5 Confirming the Streaming Replication Status.....	27
3.6 Creating Applications.....	27
3.6.1 Application Connection Server Settings.....	28
3.6.2 Notes on Indexes.....	28
3.7 Checking the Behavior.....	28
3.8 Tuning.....	28
3.8.1 Tuning to Stabilize the Database Multiplexing Mode.....	28
3.8.2 Tuning to Stabilize Queries on the Standby Server.....	28
3.8.3 Tuning to Stabilize Queries on the Standby Server (when Performing Frequent Updates on the Primary Server).....	29
3.8.4 Tuning for Optimization of Degrading Operation Using Abnormality Monitoring.....	29
3.9 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances.....	32
3.10 Backup Operation.....	33
3.10.1 Backing up Database Multiplexing Mode Information.....	33
3.10.2 Database Backup Operation.....	33
Chapter 4 Operations in Database Multiplexing Mode.....	35
4.1 Starting and Stopping Instances.....	35
4.1.1 Starting Mirroring Controller.....	35
4.1.2 Stopping Mirroring Controller.....	36
4.2 Checking the Database Multiplexing Mode Status.....	37
4.3 Manually Switching the Primary Server.....	38
4.4 Manually Disconnecting the Standby Server.....	38
4.5 Monitoring Mirroring Controller Messages.....	39
4.6 Server Maintenance.....	40
4.6.1 Rolling Updates.....	40

4.6.2 Stopping for Maintenance	45
4.7 Changes in Operation	45
4.7.1 Changes Required when the Standby Server is Stopped.....	45
4.7.2 Changing from Single Server Mode to Database Multiplexing Mode.....	46
4.7.3 Changing from Database Multiplexing Mode to Single Server Mode.....	47
4.7.4 Uninstalling in Database Multiplexing Mode.....	49
Chapter 5 Action Required when an Error Occurs in Database Multiplexing Mode.....	51
5.1 Action Required when Server Degradation Occurs.....	51
5.1.1 Operations when the Server has Started Degrading after a Switch has Occurred.....	51
5.1.1.1 Identify Cause of Error and Restore the Standby Server.....	53
5.1.1.1.1 Stop Mirroring Controller.....	53
5.1.1.1.2 Recovery of the Mirroring Controller management directory.....	54
5.1.1.1.3 Identify cause of error and perform recovery.....	54
5.1.1.2 Rebuild the Standby Server.....	55
5.1.1.3 Failback of the Primary Server.....	55
5.1.2 Operations when the Server has Started Degrading after a Disconnection has Occurred.....	56
5.1.2.1 Identify Cause of Error and Restore the Standby Server.....	57
5.1.2.1.1 Stop Mirroring Controller.....	57
5.1.2.1.2 Recovery of the Mirroring Controller management directory.....	58
5.1.2.1.3 Identify cause of error and perform recovery.....	58
5.1.2.2 Rebuild the Standby Server.....	58
5.1.3 Addressing Errors During Degrading Operation.....	58
5.2 Action Required when Automatic Switch Fails.....	59
5.3 Action Required when Automatic Disconnection Fails.....	60
5.4 Action Required when All Servers or Instances Stopped.....	60
5.5 Recovering from an Incorrect User Operation.....	63
Chapter 6 Managing Mirroring Controller Using WebAdmin.....	65
6.1 Mirroring Controller Setup.....	65
6.2 Stopping Mirroring Controller.....	67
6.3 Starting Mirroring Controller.....	67
6.4 Disabling Failover Mode.....	68
6.5 Enabling Failover Mode.....	68
6.6 Deleting Mirroring Controller Setup.....	68
6.7 Status Update after Failover.....	69
6.8 Action Required when an Error Occurs in the Combined Admin Network and Log Transfer Network.....	70
Appendix A Parameters.....	71
A.1 Parameters Set on the Primary Server.....	71
A.2 Parameters Set on the Standby Server.....	72
A.3 Network Configuration File.....	73
A.4 Server Configuration File.....	74
Appendix B Supplementary Information on Building the Primary Server and Standby Server on the Same Server.....	76
B.1 Backup Data Storage Destination Directory.....	76
B.2 Registering Service Names and Event Source Names in the Windows Service.....	76
B.3 How to Execute the mc_ctl Command.....	76
Appendix C Supplementary Procedure on Configuring for Operation in Database Multiplexing Mode.....	78
C.1 Security Policy Settings.....	78
C.2 Windows Firewall Settings.....	78
Index.....	80

Part 1 Overview of Cluster Systems

This part provides an overview of the cluster systems, and describes the types and operation modes.

Chapter 1 Definition of Cluster Operation.....	2
--	-------------------

Chapter 1 Definition of Cluster Operation

This chapter explains the features of a cluster system, the types of cluster operations, and the operation mode.

1.1 What Is a Cluster System?

A cluster system is a system that joins multiple machines (hereafter, referred to as servers) to achieve high availability and scalability throughout the system.

This allows the creation of systems that simultaneously achieve high availability and scalability.

High availability

For example, by transferring the role to another server when a server in the cluster system fails, the downtime for the entire system can be made shorter.

This type of operation mode is referred to as a "failover operation".

Scalability

The load should be distributed, in order to extend the processing capacity across the entire system by dividing multiple application processes amongst multiple servers, regardless of processing content, or to suit the processing content.

1.2 Cluster Operation Provided by FUJITSU Enterprise Postgres

In FUJITSU Enterprise Postgres, the following failover operation mode is provided in each system:

- Log shipping mode

Log shipping mode

This operation mode is 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 sending and reflecting the updated transaction log of the database from the server that receives the update (primary server) to another server (standby server).

In addition, the client driver automatically distinguishes between the primary and standby servers, so applications can be connected transparently regardless of the physical server. Accordingly, a flexible system can easily be built, for example a system that offloads connections to the standby server if the connections only perform referencing.



FUJITSU Enterprise Postgres supports only configurations comprising one standby server per primary server.



- 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.

- Refer to "[Part 2 Database Multiplexing Mode](#)" for information on the database multiplexing feature.

Part 2 Database Multiplexing Mode

This part describes database multiplexing mode.

Chapter 2 Overview of Database Multiplexing Mode.....	4
Chapter 3 Setting up Database Multiplexing Mode.....	13
Chapter 4 Operations in Database Multiplexing Mode.....	35
Chapter 5 Action Required when an Error Occurs in Database Multiplexing Mode.....	51
Chapter 6 Managing Mirroring Controller Using WebAdmin.....	65

Chapter 2 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.

2.1 What is Database Multiplexing Mode

This operation mode is 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.

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. The database will be copied in synchronous mode.

Monitoring using database multiplexing mode

- 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.

- 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. A disk error is detected when an I/O error occurs.

Failures that can be detected are those that physically affect the entire system, such as disk header or device power failures.

Figure 2.1 Failover from the primary server to the standby server

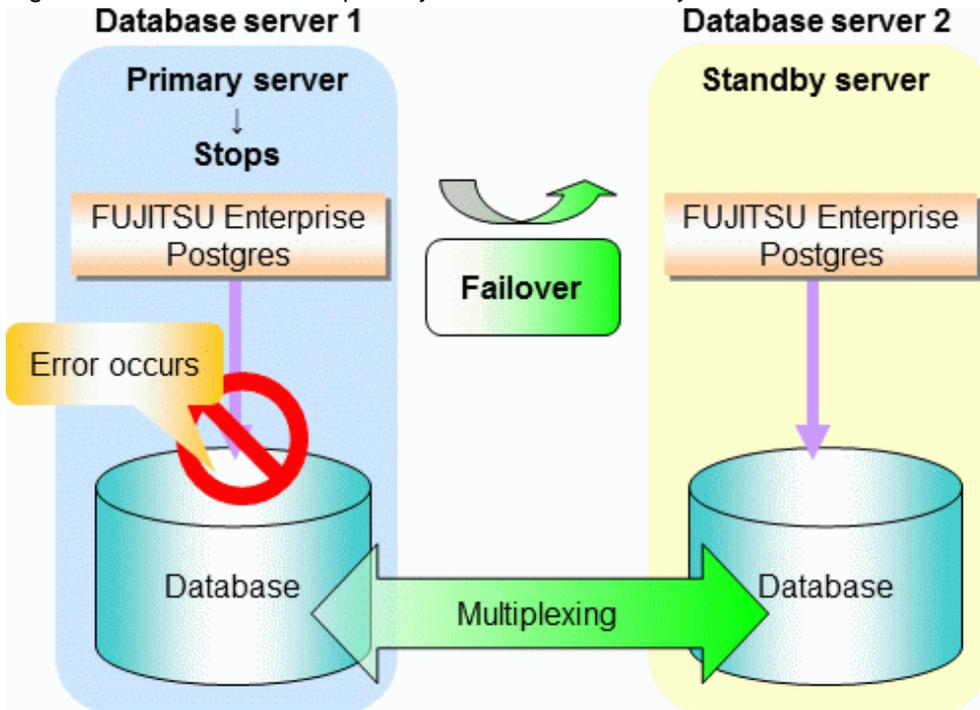
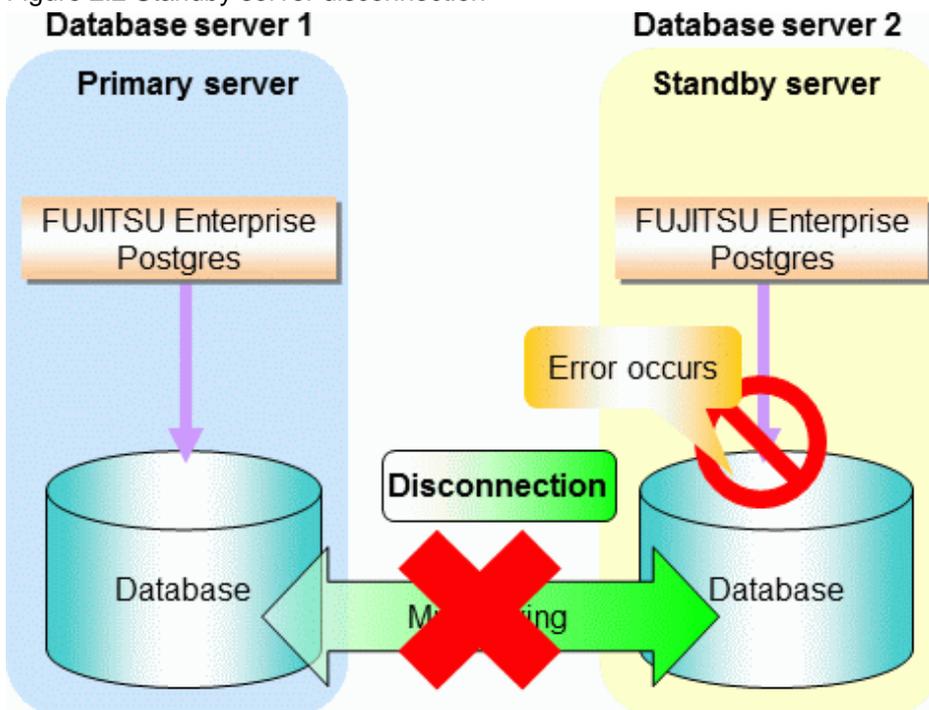


Figure 2.2 Standby server disconnection



When database multiplexing is used to perform monitoring and an error is detected, this option will automatically switch the primary server or disconnect the standby server.

Reference jobs can also be executed on the standby server.

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 "[5.1 Action Required when Server Degradation Occurs](#)" for details.

See

- Refer to "[2.2.4 Notes on Executing Reference Jobs on the Standby Server](#)" for information on reference jobs.
- The features that can be used will depend on whether automatic switch/disconnection is used. Refer to "Features that can be Installed" in the Installation and Setup Guide for Server for details.

Point

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.

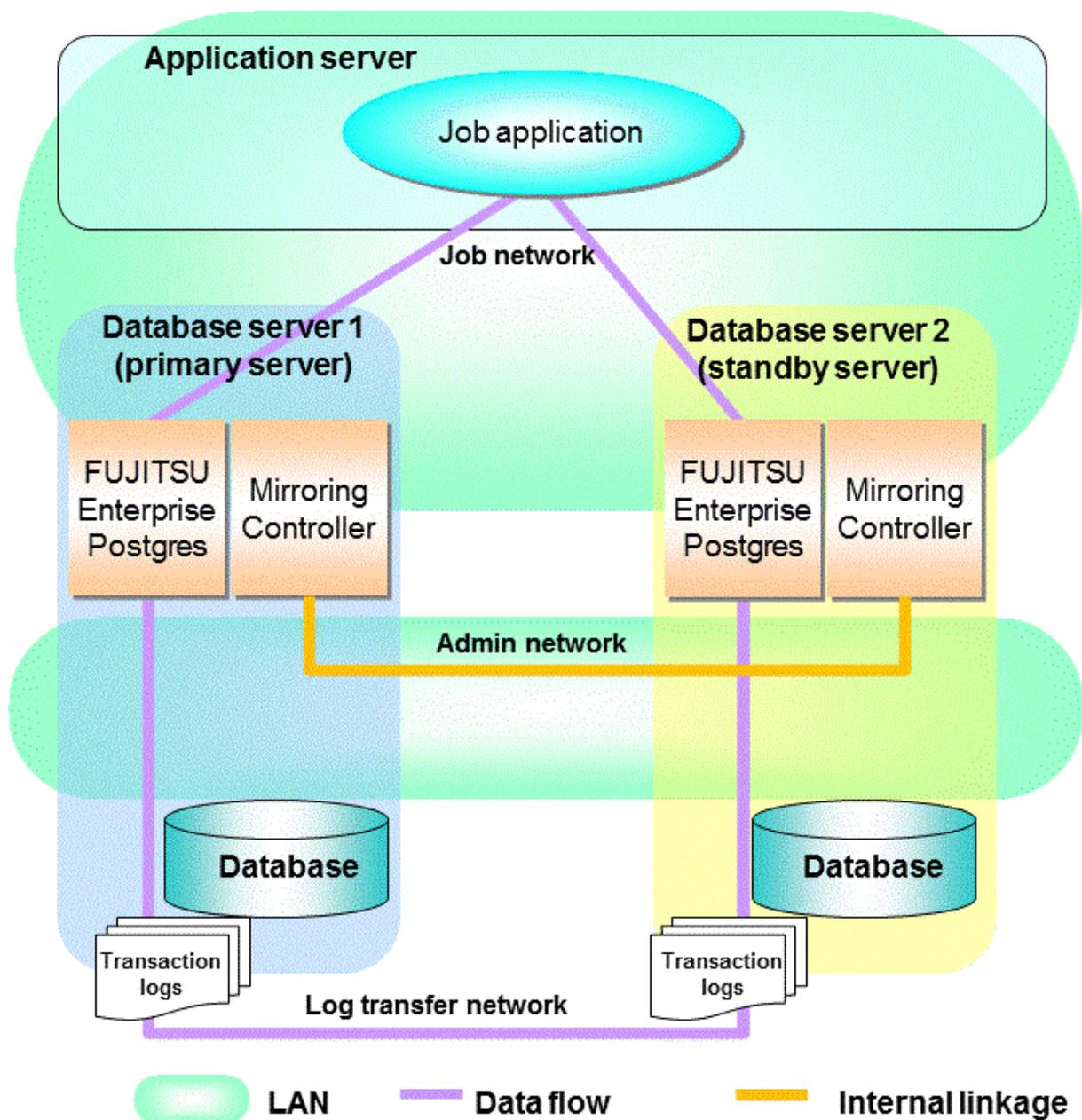
2.2 System Configuration for Database Multiplexing Mode

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.
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 2.3 System configuration for database multiplexing mode



2.2.1 Mirroring Controller 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

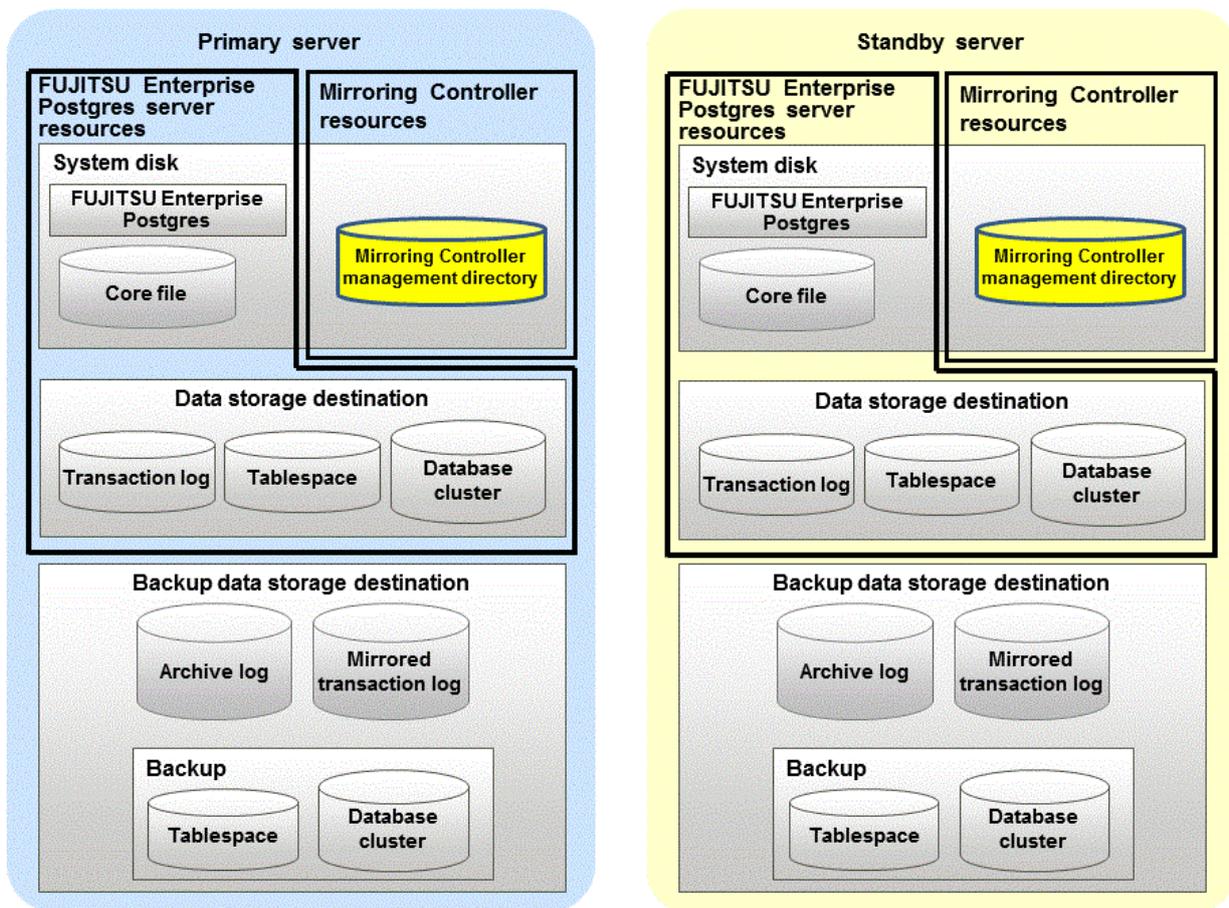
- Do not create the Mirroring Controller management directory in a directory managed by FUJITSU Enterprise Postgres, otherwise it may be deleted by mistake or may cause unexpected problems when FUJITSU Enterprise Postgres recovery is performed (such as old version of files being restored).

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.
- If the automatic switch/disconnection is enabled, do not edit synchronous_standby_names for the Mirroring Controller monitoring target instance using the methods below. Otherwise, disconnection by the Mirroring Controller may fail, since postgresql.conf settings are overridden by values specified in postgresql.auto.conf and the include_dir directives.
 - Setting the parameter using the ALTER SYSTEM statement
 - Editing the value specified in postgresql.conf or in a configuration file located in a directory listed in the include_dir directive

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 2.4 Configuration when backing up Mirroring Controller resources



2.2.2 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

In Windows Server(R) 2012 or later, the NIC teaming feature is provided by the operating system by default, and in Windows Server(R) 2008 R2 or earlier, it is provided by the network driver vendor.

2.2.3 Notes on CPU Architecture and Products

Use the same CPU architecture (endian) for the primary server and the standby server. Additionally, use the same bit architecture (32-bit/64-bit) for the FUJITSU Enterprise Postgres installation.

A server using only PostgreSQL streaming replication cannot be specified as the database multiplexing system log transfer destination.

2.2.4 Notes on Executing Reference Jobs on the Standby Server

In database multiplexing mode where automatic switch/disconnection is enabled, switch and disconnection are performed automatically. If a switch occurs when executing a reference job for the standby server, it may affect the performance of that job. The reason is that, 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 executed on the original standby server will be processed. Therefore, before executing a job using this configuration, give careful consideration to the server resource estimates, and the likely impact on performance.

Accordingly, if you are concerned about the impact on performance, pause the reference job on the original standby server, change the original primary server to the new standby server, and then resume the reference job for the new standby server.

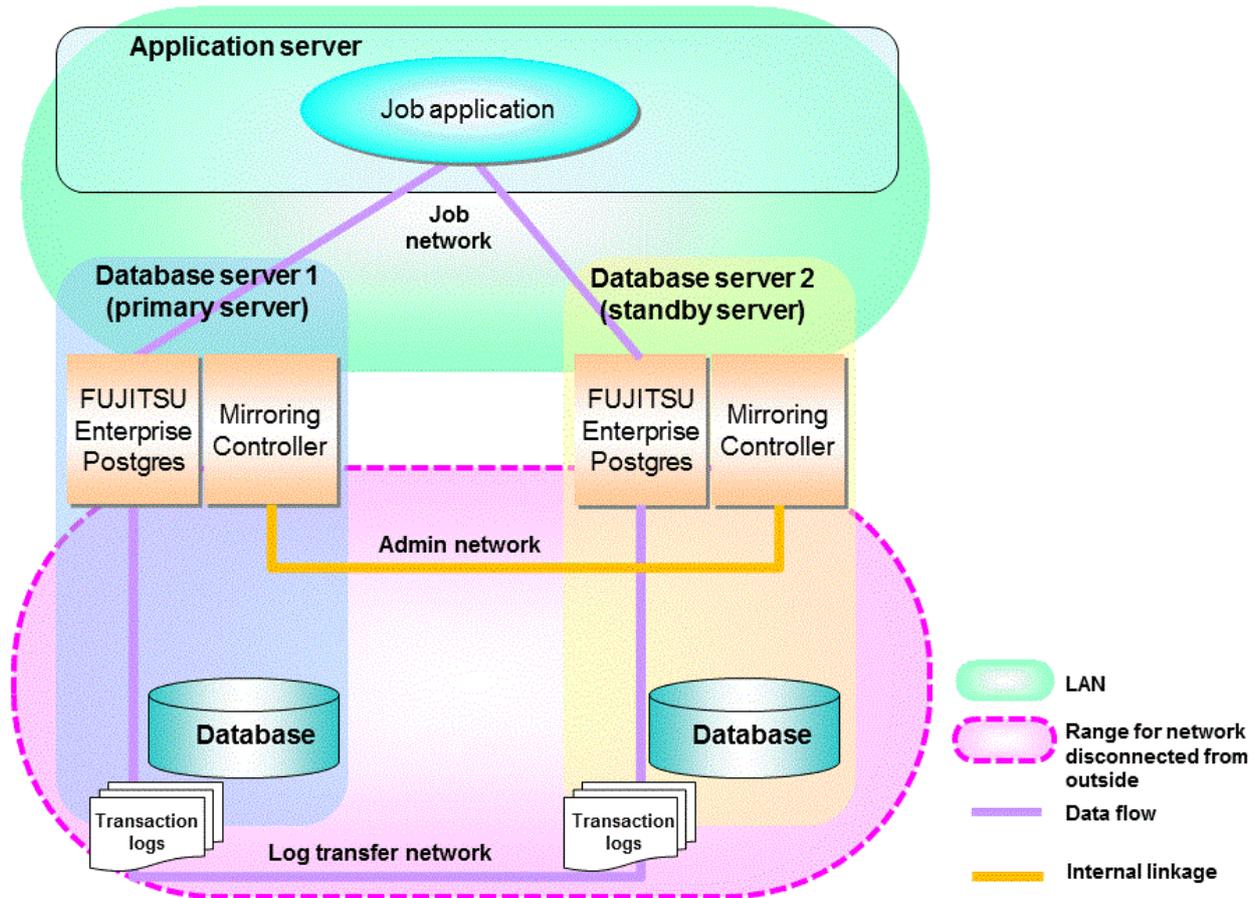
2.3 Security in Database Multiplexing

Database multiplexing mode 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 protect the confidentiality of data when transferring the transaction logs, carefully consider security and take note of the following:

- 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 below.

Figure 2.5 Security



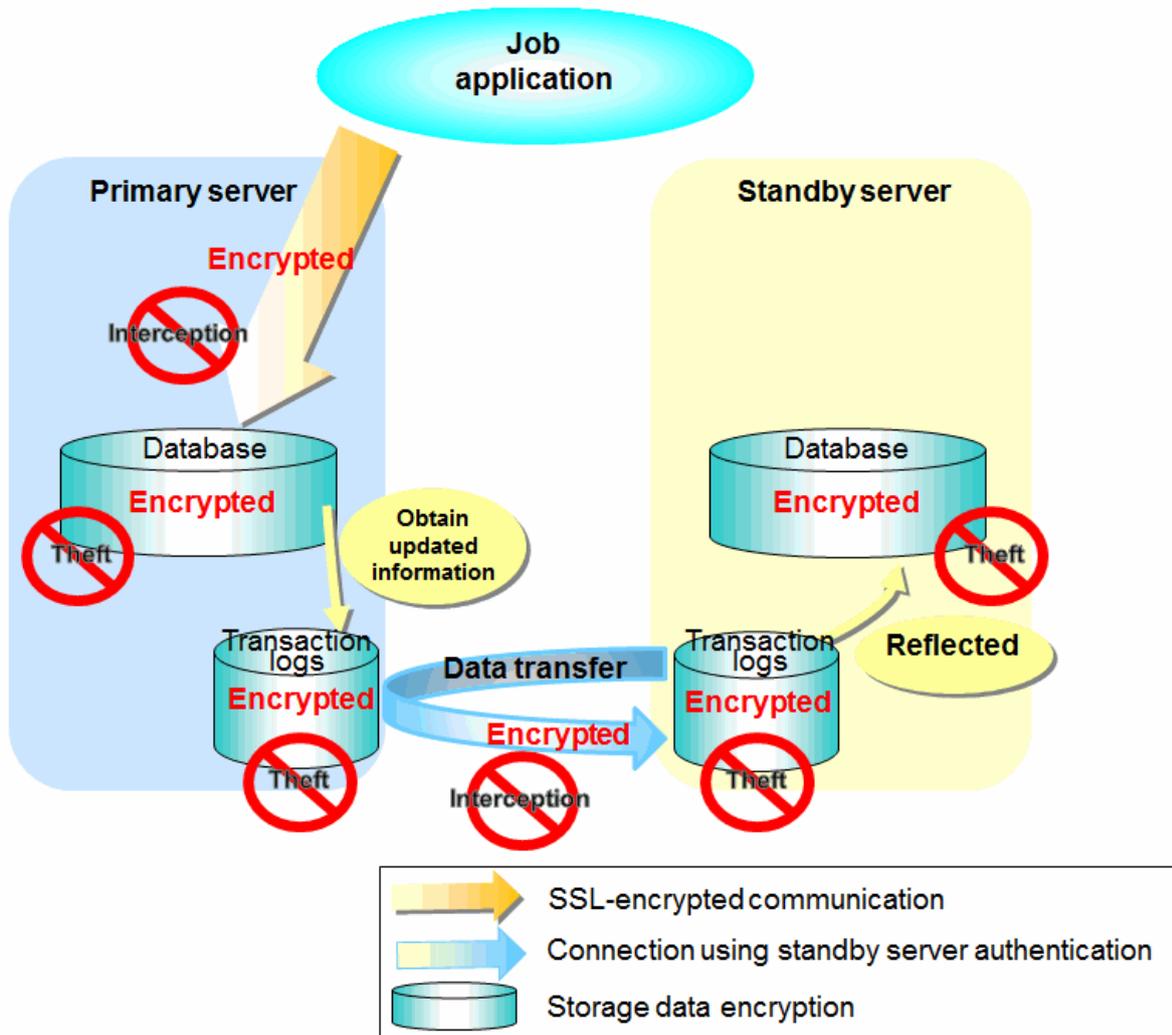
However, it may not always be possible to adopt the configuration mentioned above. For example, you may want to place the servers in a nearby/neighboring office to minimize network delays.

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.

Figure 2.6 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.

2.3.1 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`.

2.3.2 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 3 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

Setup and operations of the database multiplexing mode 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 privileges to log on as a service.
- The instance administrator user must also be registered as the database superuser.



When executing the `initdb` command, specify the `-U` option, and only perform operations in which the instance administrator user (who is the command execution user) is also a database superuser.

If an operation in which the instance administrator user is not a database superuser is performed, Mirroring Controller will be unable to connect to the instance, and therefore will not be able to perform error monitoring for the database multiplexing mode.

Matching the system times

Before starting the setup, ensure that the times in both servers 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.

Setup

Perform the procedure in the table below to set up database multiplexing mode.

Database multiplexing mode uses the streaming replication feature. If you know how to set up streaming replication, the setup of database multiplexing mode will be easy.

Step	Task		Refer to
	Primary server	Standby server	
1	Installation		3.1 Installation
2	Preparing for setup		3.2 Preparing for Setup
3	Setting up database multiplexing mode		3.3.1 Setting up Database Multiplexing Mode on the Primary Server
4	Creating, setting, and registering the instance		3.3.2 Creating, Setting, and Registering the Primary Server Instance
5		Setting up database multiplexing mode	3.4.1 Setting up Database Multiplexing Mode on the Standby Server
6		Creating, setting, and registering the instance	3.4.2 Creating, Setting, and Registering the Standby Server Instance
7	Confirming the streaming replication status		3.5 Confirming the Streaming Replication Status
8	Creating applications		3.6 Creating Applications

Step	Task		Refer to
	Primary server	Standby server	
9	Checking the behavior		3.7 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 "[4.7.2 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](#)".

3.1 Installation

Refer to the manuals of each product, and then install the product.

See

- Refer to the Installation and Setup Guide for Server for details on how to install FUJITSU Enterprise Postgres.

3.2 Preparing for Setup

This section describes the preparation required before setting up Mirroring Controller.

3.2.1 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.

3.2.2 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 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.

```
> regsvr32 "c:\Program Files\Fujitsu\fssepv<xy>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".

```
> regsvr32 /n /i:"Mirroring Controller inst1" "c:\Program Files\Fujitsu\fssepv<xy>server64\lib\mcevent.dll"
```

The parameter must be edited for each instance. Refer to "[A.4 Server Configuration File](#)" 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 an uninstall.

```
MirroringControllerOpen
```

3.2.3 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 "[C.1 Security Policy Settings](#)" for information on configuring the settings.

3.3 Setting up the Primary Server

This section explains how to set up the primary server.

3.3.1 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.



- 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.



- 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.conf file, 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.

Path of the sample file

```
install\Dir\share\mc_network.conf.sample
```

In the network.conf file, specify the host name (or IP address) and port number of the primary server and standby server, and define the network configuration that will link between the Mirroring Controller processes.

Refer to "[A.3 Network Configuration File](#)" for details.

A definition example is shown below.

In this example, the Mirroring Controller default port number 27540 is specified, and "server1" and "server2" represent any server identifiers.

Example)

```
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.

In the network.conf file, set "Read" permission 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.

Path of the sample file

```
install\Dir\share\mc_server.conf.sample
```

Set the parameters shown in the table below in the *serverIdentifier.conf* file.

Table 3.1 Parameters

Parameter	Content specified	Remarks
db_instance	'dataStorageDestinationDir'	Use ASCII characters, and specify "\\" as the path delimiter.
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.
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 to specify this parameter.
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.
heartbeat_interval	800 (milliseconds)	The time until an operation, such as the switch of the primary server, that is performed when there is no response, is calculated according to the following formula: (heartbeat_timeout (seconds) + heartbeat_interval (milliseconds)/1000) x (heartbeat_retry (count) + 1) = (1+800/1000) x (2+1) = 5.4 (seconds) According to this setting, errors are monitored at 800-millisecond intervals, and if there were three successive occurrences when there was no response for a duration of one second, the operation to switch the primary server, or disconnect the standby server from the primary server, is performed. When setting this parameter, there are some considerations to take into account to optimize degradation using abnormality monitoring. Refer to "3.8.4 Tuning for Optimization of Degrading Operation Using Abnormality Monitoring" for details.
heartbeat_timeout	1 (second)	
heartbeat_retry	2 (count)	

6. Change the access permissions for the *serverIdentifier.conf* file.

In the *serverIdentifier.conf* file, set "Read" permission 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 "[C.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:\mcdire\inst1 -w -f -P *****
```

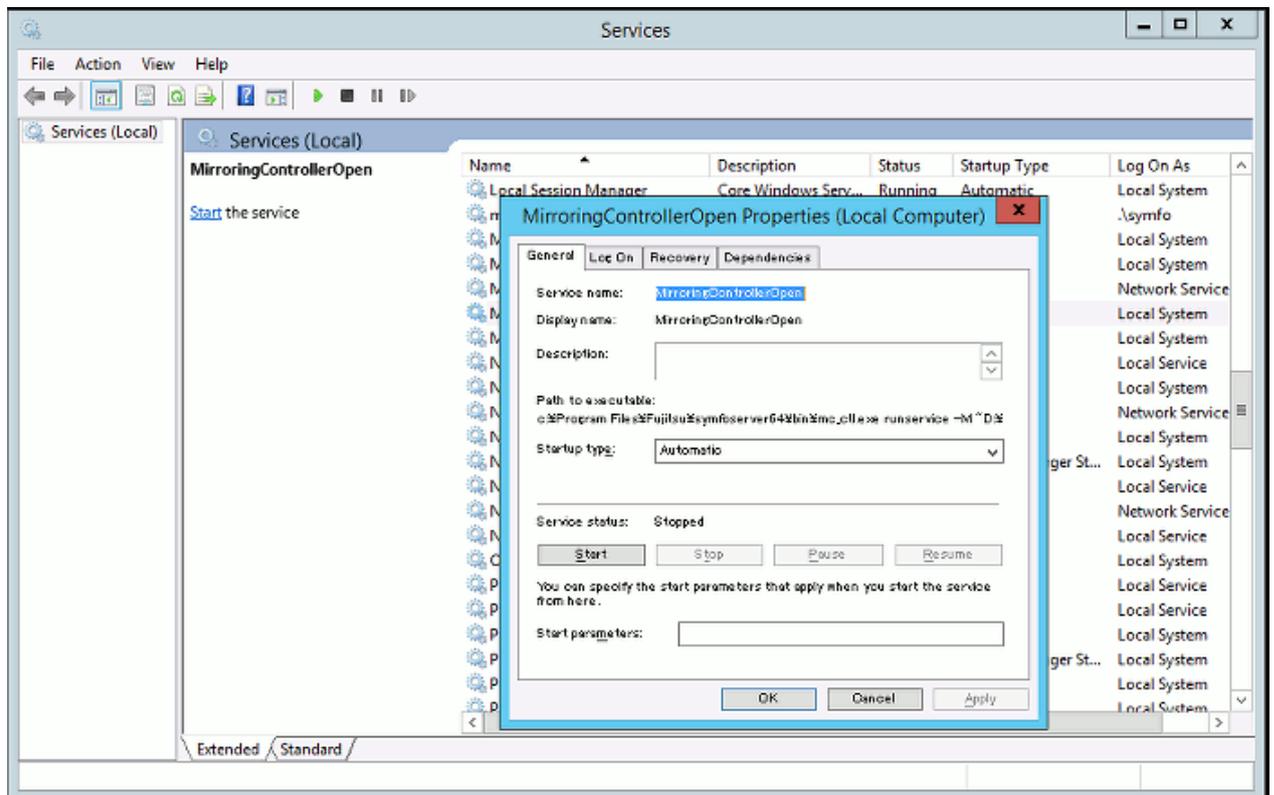
 **Note**

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 "[3.9 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 the server definition file in step 5, Mirroring Controller is registered to the Windows service as shown below.



You can execute the sc qc command to check the registration status.

3.3.2 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
 - Specify the `-w` option, so that the system waits for the start or stop process to complete

Note

Do not configure the Windows service of a multiplexed instance to perform automatic start, as it is started by Mirroring Controller.

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 "5.1.1.1.3 Identify cause of error and perform recovery" for details.

2. Configure the encryption settings for the storage data.

Create the keystore file.

Refer to "Database Multiplexing Mode" 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	<i>standbyServerAddress</i>	<i>authenticationMethod</i>
	host	replication	fsep	<i>primaryServerAddress</i>	<i>authenticationMethod</i>

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.

As the authentication method, set a 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 unknown 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. 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.

As the authentication method, set a method other than trust authentication.

- 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.

An example of setting the authentication method is shown below.

#	TYPE	DATABASE	USER	ADDRESS	METHOD
	host	postgres	fsep	127.0.0.1/32	<i>authenticationMethod</i>

Refer to the note in step 3 for notes relating to trust authentication.

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 3.2 Parameters

Parameter	Content specified	Remarks
wal_level	Literal "hot_standby"	
max_wal_senders	Number of standby servers + 1	The number of standby servers (<i>n</i>) is 1, so 2 must be specified.
synchronous_standby_names	' <i>standbyServerName</i> '	Specify the name that will identify the standby server. Enclose the name within single quotation marks ('). Do not change this parameter while Mirroring Controller is running.

Parameter	Content specified	Remarks
hot_standby	on	Specify this to execute reference jobs on the standby server. (*1)
wal_keep_segments	Number of file segments	<p>If a delay exceeding the value set in this parameter occurs, the WAL segment required later by the primary server may be deleted.</p> <p>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.</p> <p>Setting this parameter requires consideration regarding stabilization of the database multiplexing mode (refer to "3.8.1 Tuning to Stabilize the Database Multiplexing Mode" for details).</p> <p>Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on estimating the WAL segment.</p>
wal_sender_timeout	Timeout (milliseconds)	<p>Specify the time period after which it is determined that an error has occurred in the transaction log transfer on the primary server.</p> <p>By aligning this value with the value of (heartbeat_interval + heartbeat_timeout/1000) x (heartbeat_retry + 1) of the <i>serverIdentifier.conf</i> file, you can unify the time after which it is determined that an error has occurred.</p>
archive_mode	on	Specify the archive log mode.
archive_command	'cmd /c ""installDir\bin\pgx_xlogcopy.cmd" "%p" "backupDataStorageDestinationDir\archived_xlog\%f""'	Specify the command and storage destination to save the transaction log.
backup_destination	Backup data storage destination directory	<p>Specify the name of directory where to store the backup data.</p> <p>Set the privileges so that only the instance administrator user can access the specified directory.</p> <p>Specify the same full path on all servers, so that the backup data of other servers can be used to perform recovery.</p>
max_connections	The limit value for the number of standby server connections + number of simultaneous executions of mc_ctl status (*2) + 2 + number of simultaneous client connections to the instance + superuser_reserved_connections	<p>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.</p> <p>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.</p>

*1: Mandatory to use the Mirroring Controller.

*2: Number of simultaneous executions of the mc_ctl command in the status mode.

6. Start the Mirroring Controller process.

Enabling automatic switch/disconnection

As the instance administrator user, execute the `mc_ctl` command in start mode with the `-f` option specified. This 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:\mcdire\inst1 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

```
> mc_ctl start -M D:\mcdire\inst1 -w
```

7. Obtain the backup.

Use the `pgx_dmpall` command to collect the backup.

3.4 Setting up the Standby Server

This section explains how to set up the standby server.

3.4.1 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_segments` 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_segments` parameter. Additionally, setting the `wal_keep_segments` parameter requires consideration regarding stabilization of the database multiplexing mode (refer to "[3.8.1 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 "`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" permission 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" permission 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 server\identifier.conf /inheritance:r
> icacls server\identifier.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 "[C.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:\mcdire\inst1 -w -f -P *****
```

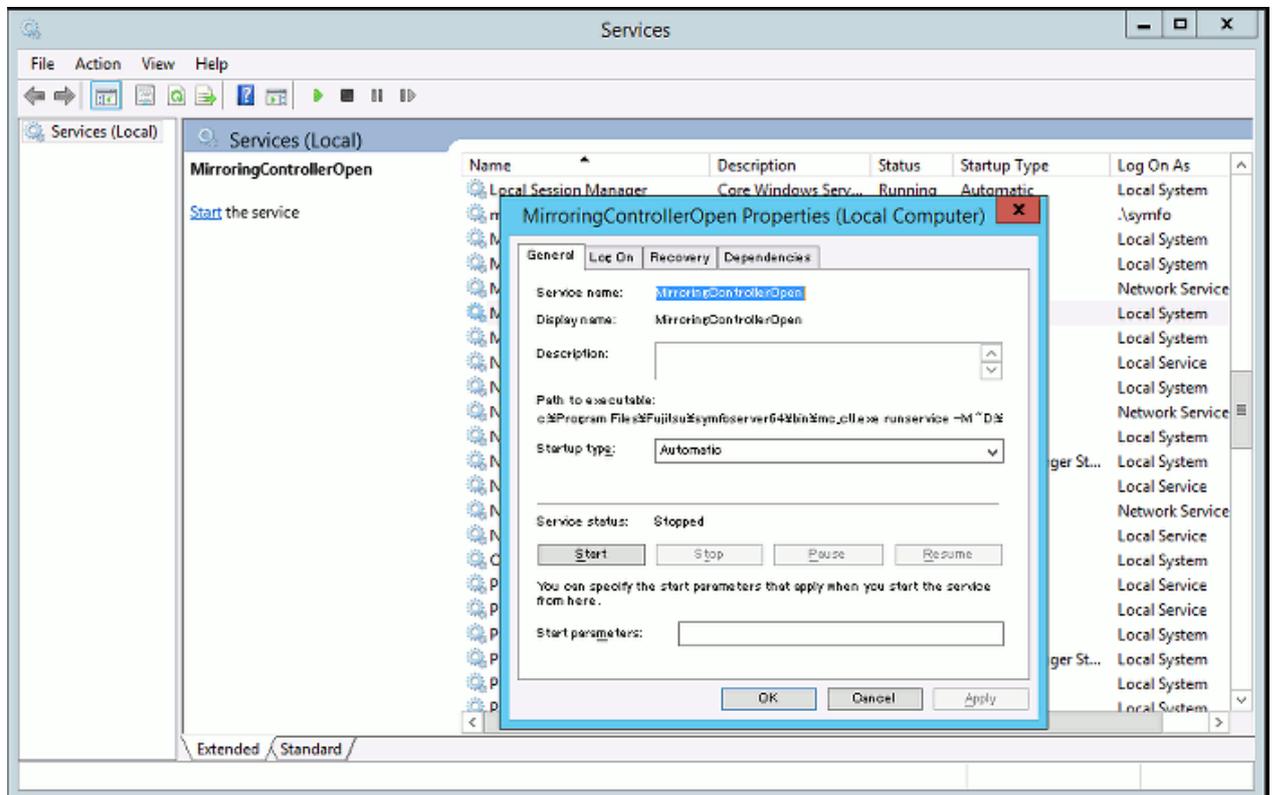
 **Note**

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 "[3.9 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 the server definition file in step 5, Mirroring Controller is registered to the Windows service as shown below.



You can execute the sc qc command to check the registration status.

3.4.2 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. Configure the encryption settings for the storage data.

Deploy a copy of the keystore file of the primary server on the standby server.

Refer to "Database Multiplexing Mode" in the Operation Guide for details.

3. Execute the `pg_basebackup` command to create a copy of the primary server instance on the standby server.

Example)

```
> pg_basebackup -D D:\mcd\inst1 --xlog --progress --verbose -R --
dbname='application_name=standbyServerName' -h primaryServerIpAddress -p primaryServerPortNumber
```

Note

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 `recovery.conf` file, enabling connections to be performed automatically.

If a password is not set in `recovery.conf`, it will be necessary to create a password file (`%APPDATA%\postgresql\pgpass.conf`), and then specify a password for the replication database.

4. Set the parameters shown in the table below in the `postgresql.conf` file.

Table 3.3 Parameters

Parameter	Content specified	Remarks
<code>synchronous_standby_names</code>	<code>'primaryServerName'</code>	Required after switching the primary server and then changing the original primary server to the new standby server. Enclose the name within single quotation marks (').

Parameter	Content specified	Remarks
		Do not change this parameter while Mirroring Controller is running.
wal_receiver_timeout	Timeout (milliseconds)	Specify the time period after which it is determined that an error has occurred in the transaction log transfer on the standby server. By aligning this value with the value of (heartbeat_interval + heartbeat_timeout) x heartbeat_retry of the <i>serverIdentifier.conf</i> file, you can unify the time after which it is determined that an error has occurred.

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
- Specify the `-w` option, so that the system waits for the start or stop process to complete

 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.

6. Start the Mirroring Controller process.

After ensuring that the Mirroring Controller process of the primary server has started, start the Mirroring Controller process.

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:\mcdire\inst1 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

```
> mc_ctl start -M D:\mcdire\inst1 -w
```

7. 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
```

3.5 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            | 14336
usesysid       | 10
username       | fsep
application_name | standby
client_addr    | 192.0.2.210
client_hostname |
client_port    | 27500
backend_start  | 2015-04-23 20:24:13.761424+09
backend_xmin   |
state          | streaming
sent_location  | 0/3000060
write_location | 0/3000060
flush_location | 0/3000060
replay_location | 0/3000060
sync_priority  | 1
sync_state     | sync
```

2. Confirm the search results of step 1.

Ensure that the connection established with the intended standby server is in synchronous mode.

Table 3.4 Items to be checked

Item	Required value
application_name	Value specified for <code>synchronous_standby_names</code> parameter in the <code>postgresql.conf</code> file of the primary server.
client_addr	IP address of the standby server.
state	"streaming".
sync_state	"sync".



- 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.

3.6 Creating Applications

This section explains how to create applications using database multiplexing, and points that should be noted when you create the applications.

3.6.1 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.

3.6.2 Notes on Indexes

Do not use hash indexes.

Even if a hash index is updated, a transaction log will not be output and the hash index will not be reflected on the standby server.

3.7 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.

3.8 Tuning

This section explains how to tune database multiplexing mode.

3.8.1 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 3.5 Parameters

Parameter	Content
wal_keep_segments	Refer to " 3.3.2 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.

3.8.2 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.

3.8.3 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.

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 "Master Server" in the PostgreSQL Documentation for details on the vacuum_defer_cleanup_age parameter.

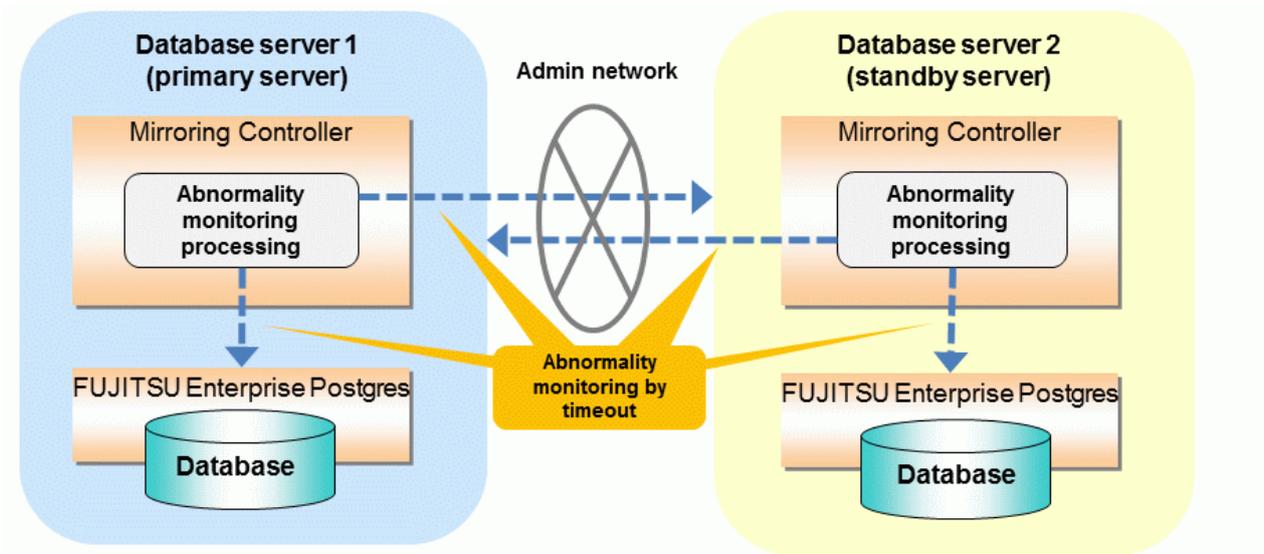
3.8.4 Tuning for Optimization of Degrading Operation Using Abnormality Monitoring

Mirroring Controller uses a monitoring method that outputs an error if the timeout time is exceeded when accessing resources targeted for monitoring. If the timeout time is short, switch/disconnection of the standby server can be performed faster, however, there is greater risk of misdetection, so an appropriate design is required.

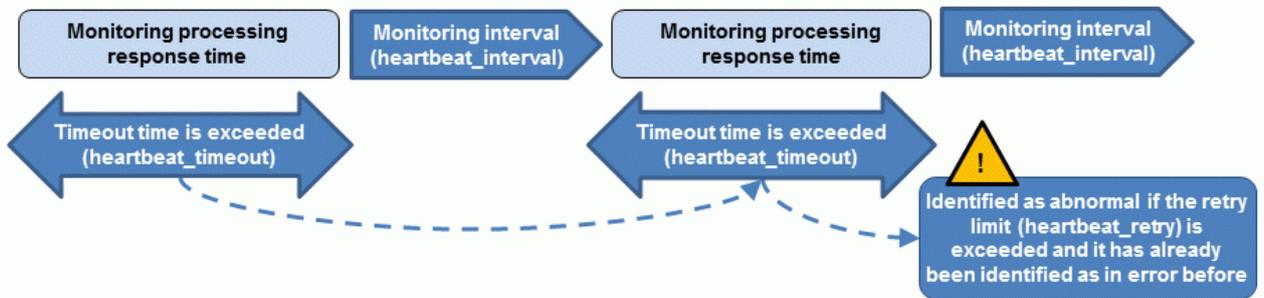
You can optimize degrading operation by editing the values for the following parameters in the server definition file in accordance with the system. Refer to "[Appendix A Parameters](#)" for information on how to edit these parameters.

Table 3.6 Parameters

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 time (heartbeat_timeout)	Set the time during which a load is placed continuously on the server or 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.
Abnormality monitoring retry times (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.)

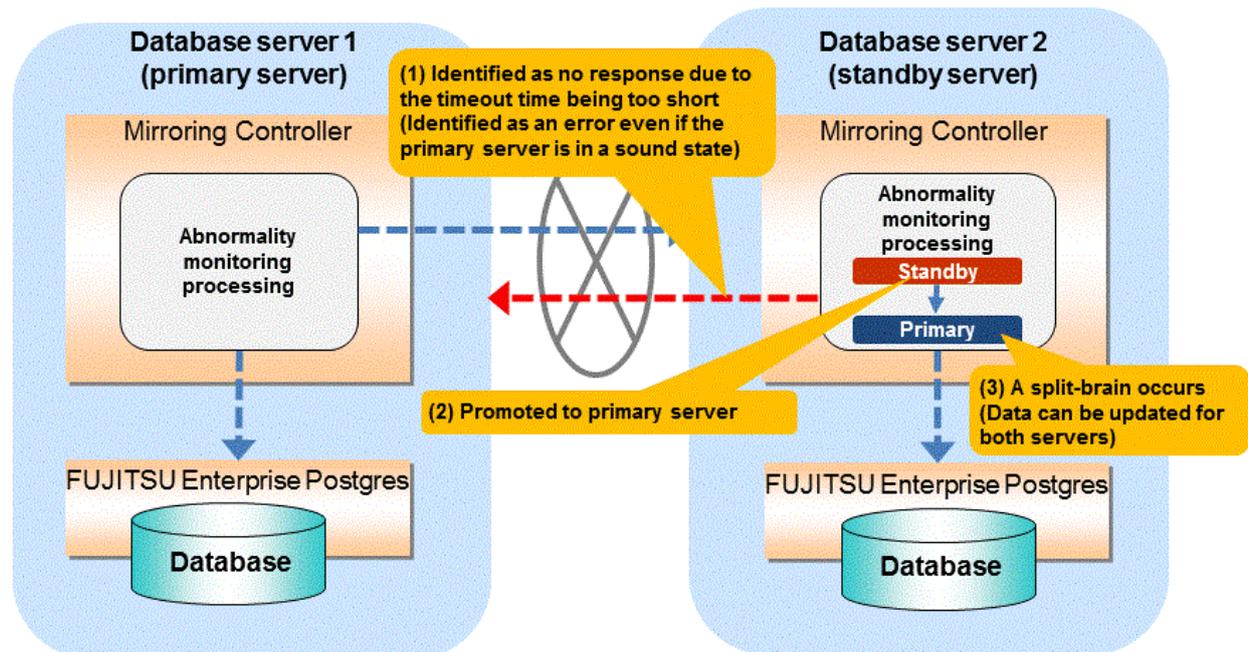


Flow of abnormality monitoring by timeout

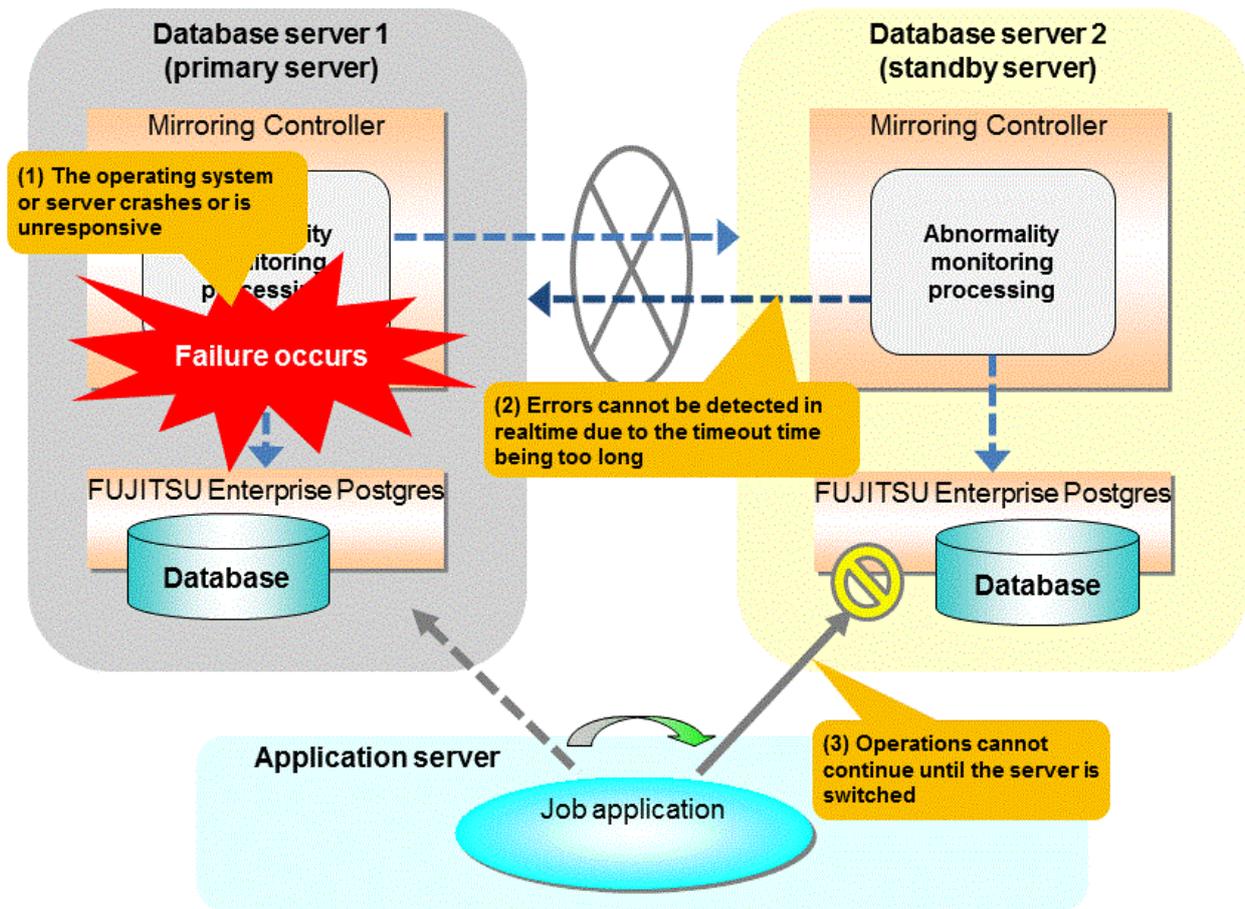


The following type of issue occurs if the tuning related to abnormality monitoring is not performed appropriately.

● If the timeout time is too short



● If the timeout time is too long



Notes regarding monitoring when the operating system or server crashes or is unresponsive

Monitoring is performed upon the aforementioned timeout when 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 event 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 "[3.4 Setting up the Standby Server](#)", build (set up) the old primary server as the new standby server, from the new primary server.

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.

3.9 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.

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 "[3.3.1 Setting up Database Multiplexing Mode on the Primary Server](#)" and "[3.4.1 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:\mcdire\inst1 -w -f -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.

3.10 Backup Operation

This section explains the backup operation for database multiplexing mode.

3.10.1 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.

3.10.2 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 5 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 4 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.

4.1 Starting and Stopping Instances

When database multiplexing mode is used, use the `mc_ctl` command or Windows service to start and stop the instance and Mirroring Controller at the same time.

Do not start or stop the instance by itself.

4.1.1 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 with the `-f` option specified.

If you start Mirroring Controller without specifying the `-f` option, 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:\mdir\inst1 -w -f
```

Disabling automatic switch/disconnection

Execute the `mc_ctl` command in start mode.

Example)

```
> mc_ctl start -M D:\mdir\inst1 -w
```



Point

To start the Mirroring Controller process only, execute the `mc_ctl` command in start mode with the `--mc-only` option specified.



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 "3.9 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for details.



Note

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 "2.1 What is Database Multiplexing Mode" will not be detected until Mirroring Controller is started.

4.1.2 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:\mdir\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 [Administrativehpq Tools], then [Services] to open the [Services] window, and then select the Mirroring Controller service and click the [Stop] menu.

Note

- 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.

4.2 Checking the Database Multiplexing Mode Status

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.

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:\mcdire\inst1

mirroring status
-----
switchable
server_id  host_role      host           host_status  db_proc_status  disk_status
-----
nd1        primary        192.0.2.100   normal       normal          normal
nd2        standby        192.0.2.110   normal       normal          normal
```

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_location`, `replay_location`) match, and that they are fully synchronized.

Example)

```
postgres=# select * from pg_stat_replication;
-[ RECORD 1 ]-----+-----
pid           | 14336
usesysid      | 10
username      | fsep
application_name | standby
```

client_addr		192.0.2.210
client_hostname		
client_port		27500
backend_start		2015-04-23 20:24:13.761424+09
backend_xmin		
state		streaming
sent_location		0/3000060
write_location		0/3000060
flush_location		0/3000060
replay_location		0/3000060
sync_priority		1
sync_state		sync



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.

4.3 Manually Switching the Primary Server

The primary server cannot be switched automatically in the following case:

- If automatic switch/disconnection is disabled

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:\mcdire\inst1
```

4.4 Manually Disconnecting the Standby Server

The standby server cannot be disconnected automatically in the following case:

- If automatic switch/disconnection is disabled

In this case, to manually disconnect the standby server, execute the mc_ctl command in stop mode on the standby server and comment out the synchronous_standby_names parameter in the postgresql.conf file on the primary server.

1. Execute the mc_ctl in stop mode on the standby server.

Example)

```
> mc_ctl stop -M D:\mcdire\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
```

4.5 Monitoring Mirroring Controller Messages

The messages that are output by Mirroring Controller are output to the event log. If the automatic switch fails, for example, an important message related to the continuation of the operation may be output, so ensure that the event log messages are monitored.

Point

To monitor message types considered to be important, a setting must be configured in the event log beforehand.

Refer to the operating system manual, check if the monitoring target message type message is output to the event log, and configure the setting if required.

Display format

```
eventSourceName[processId]: messageType: messageContent (messageNumber)
```

Specify the event source name in the event_source parameter of the serverIdentifier.conf file.

The message types output by Mirroring Controller, their severity, and their corresponding value in the event log are shown in the table below.

Table 4.1 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)	
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 one of the multiplexed database systems requiring recovery of this 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 this 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.

4.6 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.

4.6.1 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.

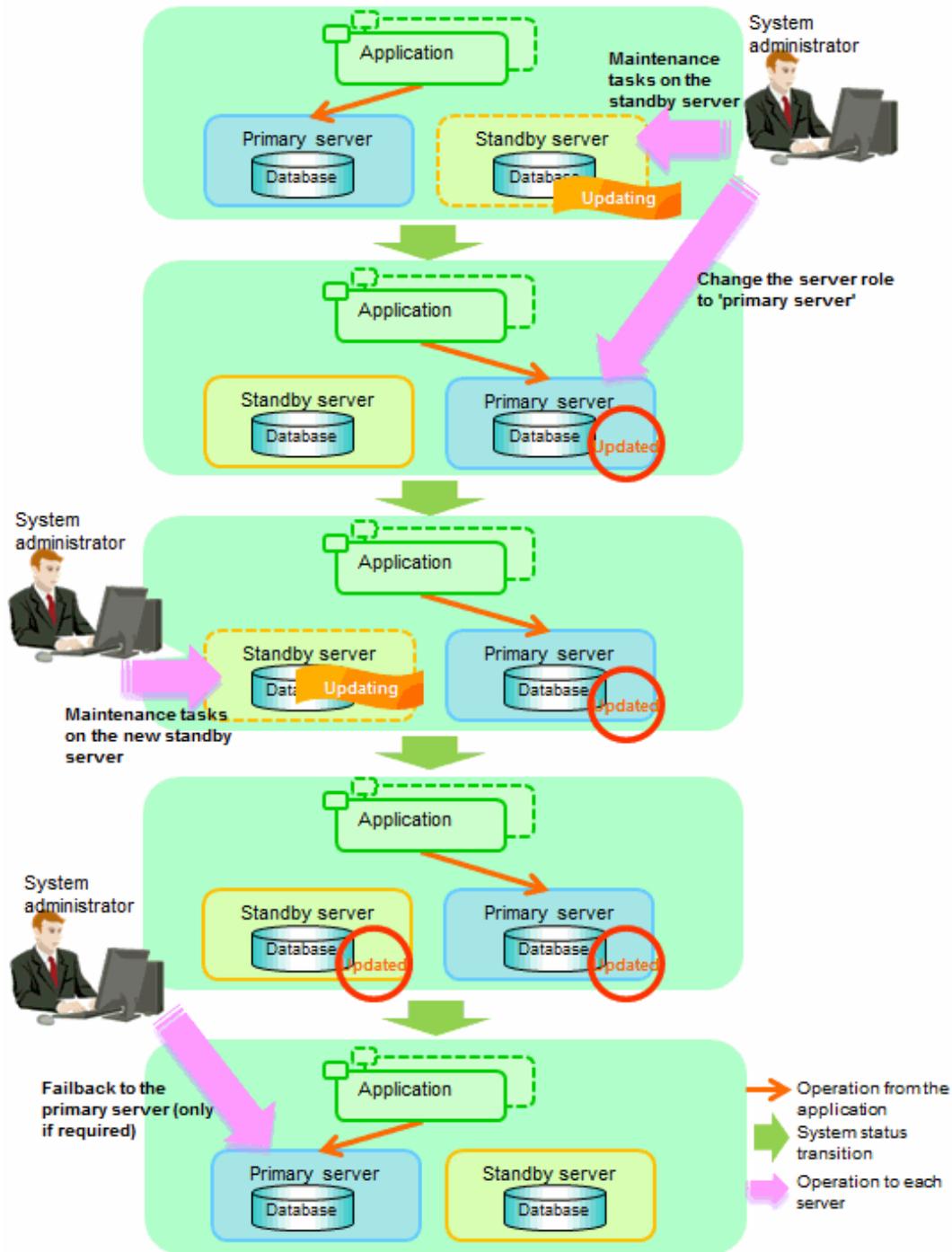


See

.....
If the downtime due to the maintenance of the standby server is expected to be long, refer to "Standby server downtime" in "[4.7.1 Changes Required when the Standby Server is Stopped](#)".
.....

The flow of a rolling update is shown below.

Figure 4.1 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:\mcdire\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 "3.9 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:\mcd\inst1 --xlog --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 "3.4.2 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 with the `-f` option specified. This enables automatic switch and disconnection.

If you start Mirroring Controller and the instance without specifying the `-f` option, automatic switch and 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:\mcdire\inst1 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

```
> mc_ctl start -M D:\mcdire\inst1 -w
```

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:\mcdire\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:\mcdire\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 "[3.9 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:\mcdir\inst1 --xlog --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 "[3.4.2 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 with the -w and -f options specified. This enables automatic switch and disconnection.

If you start Mirroring Controller and the instance without specifying the -f option, automatic switch and 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:\mcdire\inst1 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode with the -w option specified.

Example)

```
> mc_ctl start -M D:\mcdire\inst1 -w
```

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 "5.1.1.3 Failback of the Primary Server" for details.



Obtain a backup as soon as this task is complete.

4.6.2 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.

Example)

```
> mc_ctl stop -M D:\mcdire\inst1 -a
```

4.7 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

4.7.1 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, 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 standby server event log, and all transaction logs that should be transferred to the standby server may be lost.

- The `wal_keep_segments` 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 "[3.8.1 Tuning to Stabilize the Database Multiplexing Mode](#)" for details.



Note

.....
The standby server must be rebuilt in both cases above.

Take the action advised in the recovery operation that starts from "[5.1.1.1.3 Identify cause of error and perform recovery](#)" through to "[5.1.1.2 Rebuild the Standby Server](#)".
.....

4.7.2 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 3 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 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.

2. Stop the application jobs

Stop the application jobs to be connected to the primary server.

3. 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 "[3.3.2 Creating, Setting, and Registering the Primary Server Instance](#)" on the primary server.

4. Set up database multiplexing mode on the standby server

Refer to "[3.4.1 Setting up Database Multiplexing Mode on the Standby Server](#)" for details.

5. Create the standby server instance and start it

Refer to "[3.4.2 Creating, Setting, and Registering the Standby Server Instance](#)" for details.

After the above steps are completed, refer to the remaining explanations in "[Chapter 3 Setting up Database Multiplexing Mode](#)" and ensure that the required settings and operations are completed.

4.7.3 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.

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:\mcdire\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:\mcdire\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:\mcdire\inst1
```

4. Delete registries related to the event log

If error logs are output to the event log in "3.2.2 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\fssepv<xy>server64\lib\mcevent.dll"
```

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:\mcdire\inst1
```



Refer "Security-Related Notes" in the Operation Guide for information on deleting the data securely.

6. 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:\mcdire\inst1
```

7. 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:\mcdire\inst1
```

8. Delete registries related to the event log on the primary server

If error logs are output to the event log in "3.2.2 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\fssepv<xy>server64\lib\mcevent.dll"
```

9. Delete the database multiplexing mode settings that were configured for the primary server instance.

Take the appropriate action for each parameter and resource by editing postgresql.conf as shown below. Also, delete recovery.conf or change its name to recovery.done for example.

File	Parameter	Action
postgresql.conf	wal_level	Delete the "hot_standby" value that was specified.
	max_wal_senders	Revert the value to the one set before database multiplexing mode was set.
	synchronous_standby_names	Delete.

File	Parameter	Action
	wal_sender_timeout	
	listen_addresses	Revert the value to the one set before database multiplexing mode was set.
	max_connections	
	synchronous_standby_names	Delete.
	hot_standby	

- Mirroring Controller management directory

Additionally, 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

After these actions are performed, ensure that the backup data is collected when starting the single operation.



- Refer to "Security-Related Notes" in the Operation Guide for details on deleting the data securely.
- Refer to "3.10 Backup Operation" for details on the backup operation.
- Refer to "Appendix A Parameters" for details on the postgresql.conf file parameters.

4.7.4 Uninstalling in Database Multiplexing Mode

This section explains how to uninstall FUJITSU Enterprise Postgres on a server using database multiplexing mode.

You must delete the Mirroring Controller settings from both the primary server and standby server before uninstalling FUJITSU Enterprise Postgres.

1. Stop the multiplexed instances and Mirroring Controller

Refer to "4.1 Starting and Stopping Instances" 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:\mcdire\inst1
```

3. Delete registries related to the event log

If messages are output to the event log, DLLs are registered in accordance with "3.2.2 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.

```
> regsvr32 /u "c:\Program Files\Fujitsu\fspev<xy>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.

```
> regsvr32 /u /i:"Mirroring Controller inst1" "c:\Program Files\Fujitsu\fsepv<XY>server64\lib\ncevent.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 "[3.2.2 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.

Chapter 5 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

5.1 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:\mcdire\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:\mcdire\inst1
mirroring status
-----
not-switchable
:
```



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 "5.1.3 Addressing Errors During Degrading Operation", and then recover to database multiplexing mode.

5.1.1 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

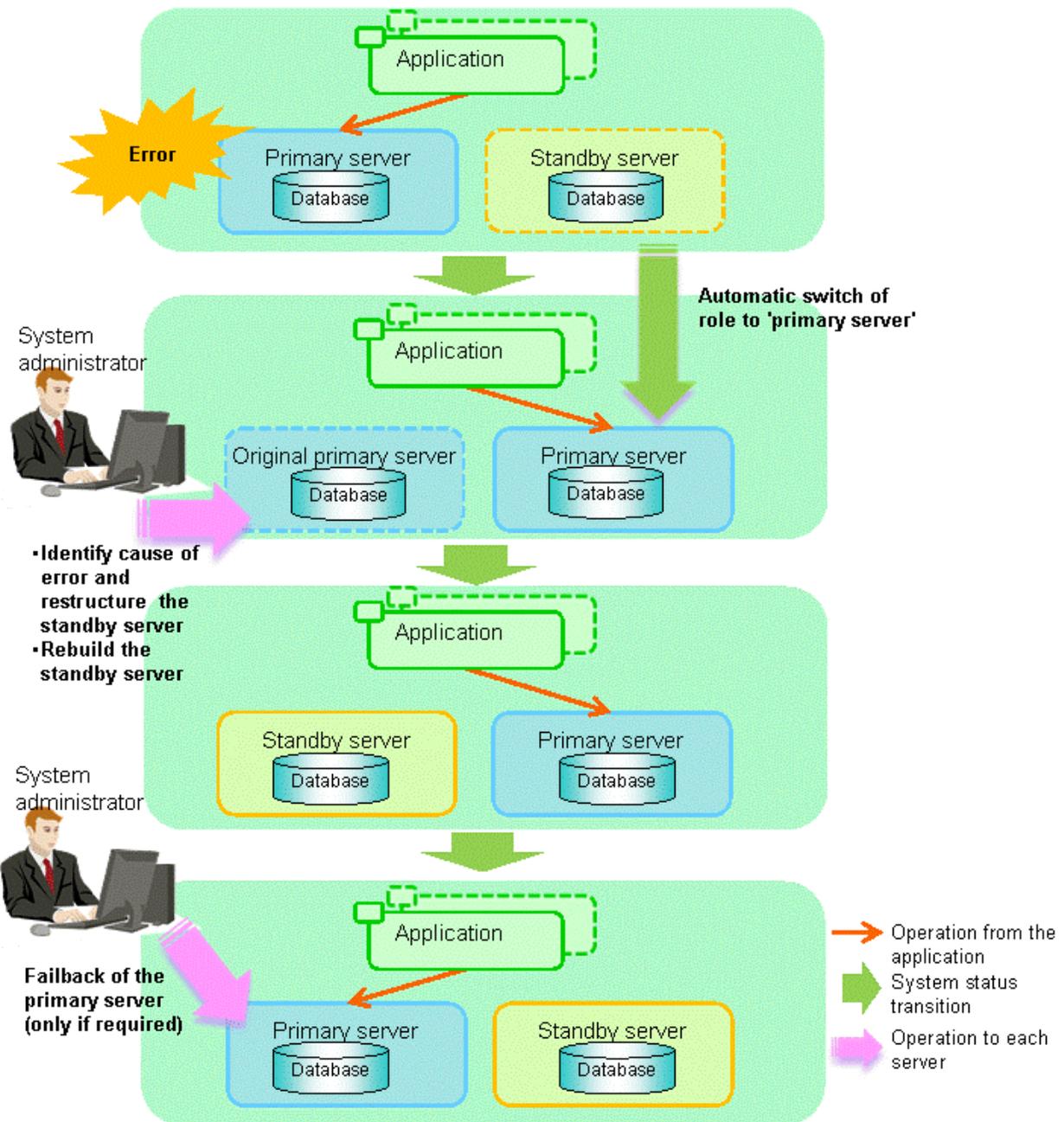
- After a switch has occurred as a result of an abnormality on the primary server, the database will not have a multiplexed configuration until the standby server is rebuilt. Remove the cause of the error as quickly as possible, and then rebuild the standby server.
 - If the reference job was executed 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 5.1 Flow of operations



5.1.1.1 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](#)

5.1.1.1.1 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:\mcdire\inst1
```

This also stops the instance that is required to perform the recovery.



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.

5.1.1.1.2 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.

5.1.1.1.3 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:

- `pg_basebackup`

Creates a copy of all resources of the primary server instance. This command is used for example when creating the first instance on the standby server.

- `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 and the original primary server was stopped normally, or
- b. The original primary server was stopped normally and 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.

If it is required to start the original primary server again to then stop it normally, both servers will temporarily operate as primary servers. If data is updated on the original primary server, data corruption will occur, so take measures such as changing settings in `pg_hba.conf` to disallow client connections.



Refer to "pg_rewind" under "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. Use the `pg_ctl` command to start the original primary server instance.

Example)

```
> pg_ctl start -w -D D:\mcdire\inst1
```

2. Use the `pg_ctl` command to stop the original primary server instance.

Example)

```
> pg_ctl stop -D D:\mcdire\inst1
```

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 db1 --source-server='user=userName host=newPrimaryServerHostName
port=newPrimaryServerPortNumber'
```

5.1.1.2 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 with the `-f` option specified. This 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:\mcdire\inst1 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

```
> mc_ctl start -M D:\mcdire\inst1 -w
```

5.1.1.3 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:\mcdire\inst1
```

After executing the `mc_ctl` command in switch mode, the status will be as follows:

Example)

```
> mc_ctl status -M D:\mcdire\inst1
mirroring status
-----
switched
server_id  host_role                host          host_status  db_proc_status  disk_status
-----
nd1        primary                  192.0.2.100  normal       abnormal(postmaster) normal
nd2        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:\mcdire\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:\mcdire\inst1 --xlog --progress --verbose -R --  
dbname='application_name=standbyServerName' -h primaryServerHostName -p primaryServerPortNumber
```



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 "3.4.2 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 with the `-w` and `-f` options specified. This 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:\mcdire\inst1 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the `mc_ctl` command in start mode with the `-w` option specified.

Example)

```
> mc_ctl start -M D:\mcdire\inst1 -w
```

5.1.2 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.



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.

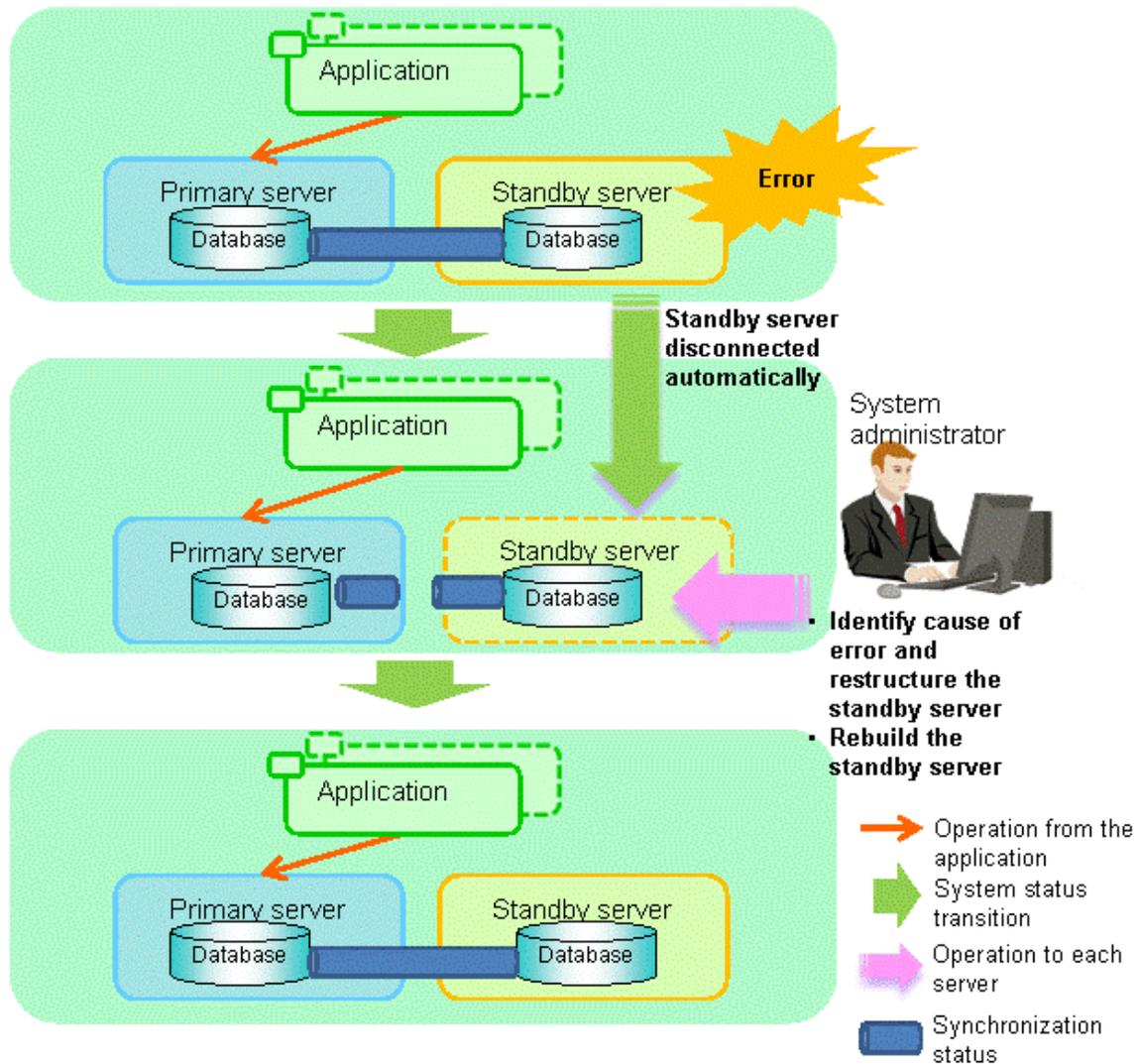
If the disconnection occurred and the server has started degrading, perform the following operations to recover the standby server and revert it to its original state:

- [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 5.2 Flow of operations



5.1.2.1 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](#)

5.1.2.1.1 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:\mcdire\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.

5.1.2.1.2 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.

5.1.2.1.3 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:\mcdir\inst1 --xlog --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 "3.4.2 Creating, Setting, and Registering the Standby Server Instance", and then perform the recovery.

5.1.2.2 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 with the `-f` option specified. This 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 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

```
> mc_ctl start -M D:\mcdir\inst1 -w
```

5.1.3 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 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 "[4.1.1 Starting Mirroring Controller](#)" and "[4.1.2 Stopping Mirroring Controller](#)" for information on restarting Mirroring Controller.

5.2 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:

- Execute the `mc_ctl` command in switch mode

Execute the `mc_ctl` command in switch mode on the primary server or the standby server.

Example)

```
> mc_ctl switch -M D:\mcdire\inst1
```



Note

If the server fails during execution of the `mc_ctl` command, automatic switch, or automatic disconnection, or if communication between the primary server and the standby server is lost, a message indicating that the `mc_ctl` command is being executed may be output even though the command is not running elsewhere, and the command may end abnormally.

In this case, switch the server in accordance with the "Execute the `mc_ctl` command in stop mode" procedure below.

- Execute the `mc_ctl` command in stop mode

Execute the `mc_ctl` command in stop mode on the server on which the error occurred.

Example)

```
> mc_ctl stop -M D:\mcdire\inst1
```

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 `mc_ctl` command to forcibly stop Mirroring Controller.

Example)

```
> mc_ctl stop -M D:\mcdire\inst1 -e
```



See

Recovery to database multiplexing mode

Refer to "5.1.1.2 Rebuild the Standby Server" and "5.1.1.3 Failback of the Primary Server" for information on recovery to database multiplexing mode.

5.3 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.

In this case, perform the following procedure to disconnect the primary server manually.

1. On the standby server, execute the mc_ctl command in stop mode with the -e option specified.

Example)

```
> mc_ctl stop -M D:\mcdire\inst1 -e
```

2. On the primary server, comment out the synchronous_standby_names parameter in the postgresql.conf file.

Example)

```
:  
# synchronous_standby_names = 'standby1 '
```

3. Execute the pg_ctl command in reload mode on the primary server.

Example)

```
> pg_ctl reload -D D:\database\inst1
```

 See

Refer to "pg_ctl" in Reference for information on the command.

 See

Recovery to database multiplexing mode

Refer to "5.1.2.2 Rebuild the Standby Server" for information on recovery to database multiplexing mode.

5.4 Action Required when All Servers or Instances Stopped

This section explains what happens when all servers or instances on the server have stopped, so jobs cannot continue.

 See

Recovery to database multiplexing mode

Refer to "5.1.1.2 Rebuild the Standby Server" and "5.1.1.3 Failback of the Primary Server" for information on recovery to database multiplexing mode.

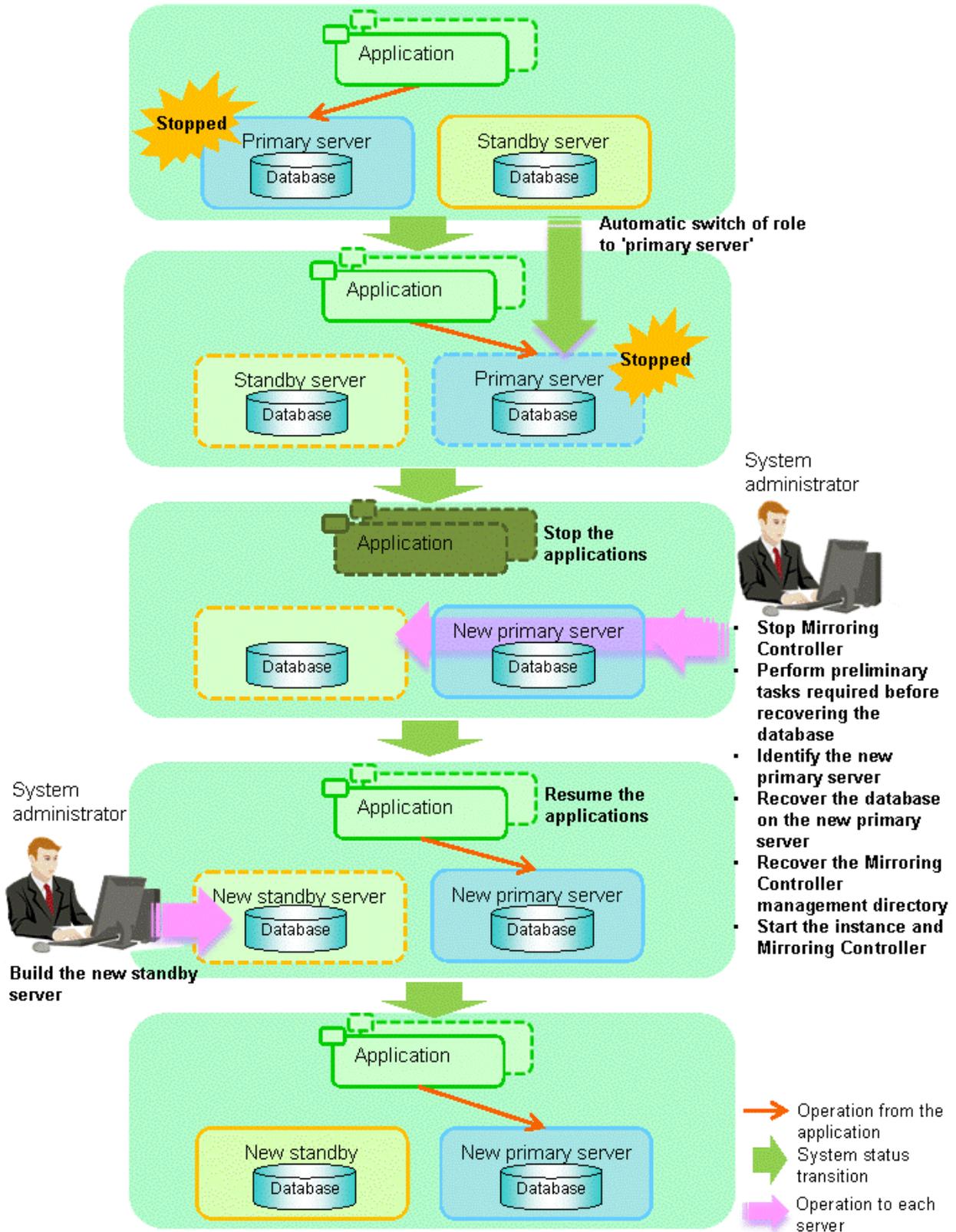
Overview of recovery operations

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 5.3 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:\mcdire\inst1
```



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:\mcdire\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>\*_xlog
```

- 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>\*_xlog
```

- c. Execute the `pgx_rcvall` command on the new primary server, specifying the backup storage destination directory you used in step b.

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
```



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 with the `-f` option specified. This 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:\mcdire\inst1 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

```
> mc_ctl start -M D:\mcdire\inst1 -w
```

8. Resume applications

Resume execution of applications.

9. Build the new standby server

Refer to "[3.4 Setting up the Standby Server](#)" for information on building (setting up) a standby server from the primary server.



It is not necessary to repeat steps that have already been performed, such as registering to Windows services.

5.5 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:\mcdire\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 with the -f option specified. This 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:\mcdire\inst1 -w -f
```

Disabling automatic switch/disconnection

As the instance administrator user, execute the mc_ctl command in start mode.

Example)

```
> mc_ctl start -M D:\mcdire\inst1 -w
```

6. Build the new standby server

Refer to "3.4 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 6 Managing Mirroring Controller Using WebAdmin

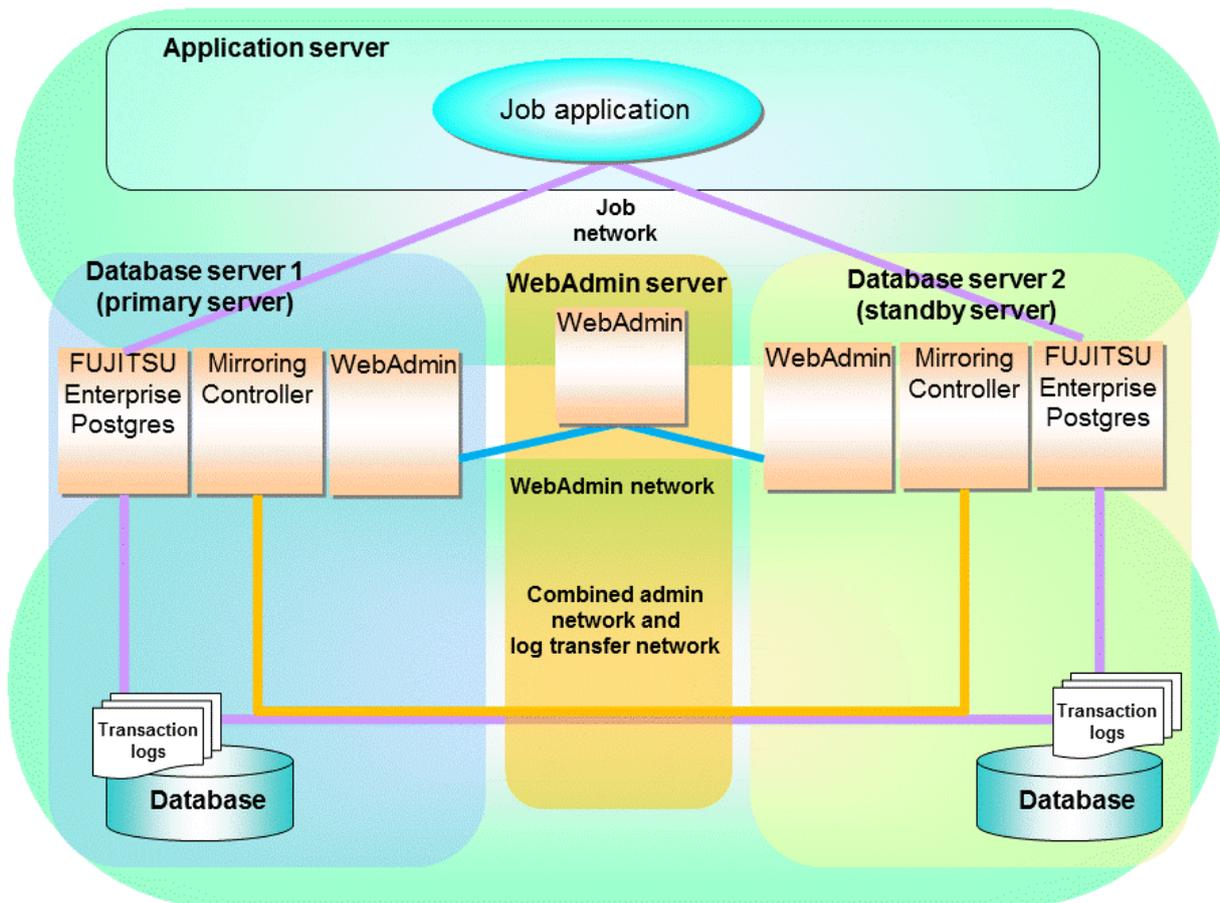
This chapter describes how to set up and manage Mirroring Controller in a streaming replication cluster using WebAdmin.

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 having master and synchronous standby instances.

The configuration of the database multiplexing system built using WebAdmin is shown below:

Figure 6.1 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].

6.1 Mirroring Controller Setup

Perform the following procedure to setup Mirroring Controller in a streaming replication cluster. The option for this setup is accessible only in synchronous standby instance.

1. In the [Instances] tab, select the synchronous standby instance on which Mirroring Controller needs to be set up.
2. Click .

3. Enter the information for the Mirroring Controller to be setup.

In the example below, Mirroring Controller is being setup for the replication cluster having master instance "inst1" and standby instance "inst1s".

The screenshot displays the 'Mirroring Controller setup' page in the Fujitsu Enterprise Postgres administration console. The page is divided into two main sections: 'Mirroring Controller setup for master instance' and 'Mirroring Controller setup for standby instance'. Each section contains a table of configuration parameters. The 'Enable automatic switch over' option is set to 'Yes'. The master instance (inst1) is configured with port 27500 and localhost. The standby instance (inst1s) is configured with port 27502 and host FASTDEVVMS. Both instances share the same Mirroring Controller port (27505) and management directory (c:\temp\mcdir\inst1/inst1s). Heartbeat settings are consistent for both: 800ms interval, 1s timeout, and 2 retries.

Mirroring Controller setup for master instance	
Instance name	inst1
Instance port	27500
Instance host	localhost
Mirroring Controller port *	27505
Mirroring Controller management directory *	c:\temp\mcdir\inst1
Heartbeat interval (in milliseconds) *	800
Heartbeat timeout (in seconds) *	1
Heartbeat retry *	2

Mirroring Controller setup for standby instance	
Instance name	inst1s
Instance port	27502
Instance host	FASTDEVVMS
Mirroring Controller port *	27505
Mirroring Controller management directory *	c:\temp\mcdir\inst1s
Heartbeat interval (in milliseconds) *	800
Heartbeat timeout (in seconds) *	1
Heartbeat retry *	2

The instance name, host address and port of the master and standby instances are displayed for easy reference.

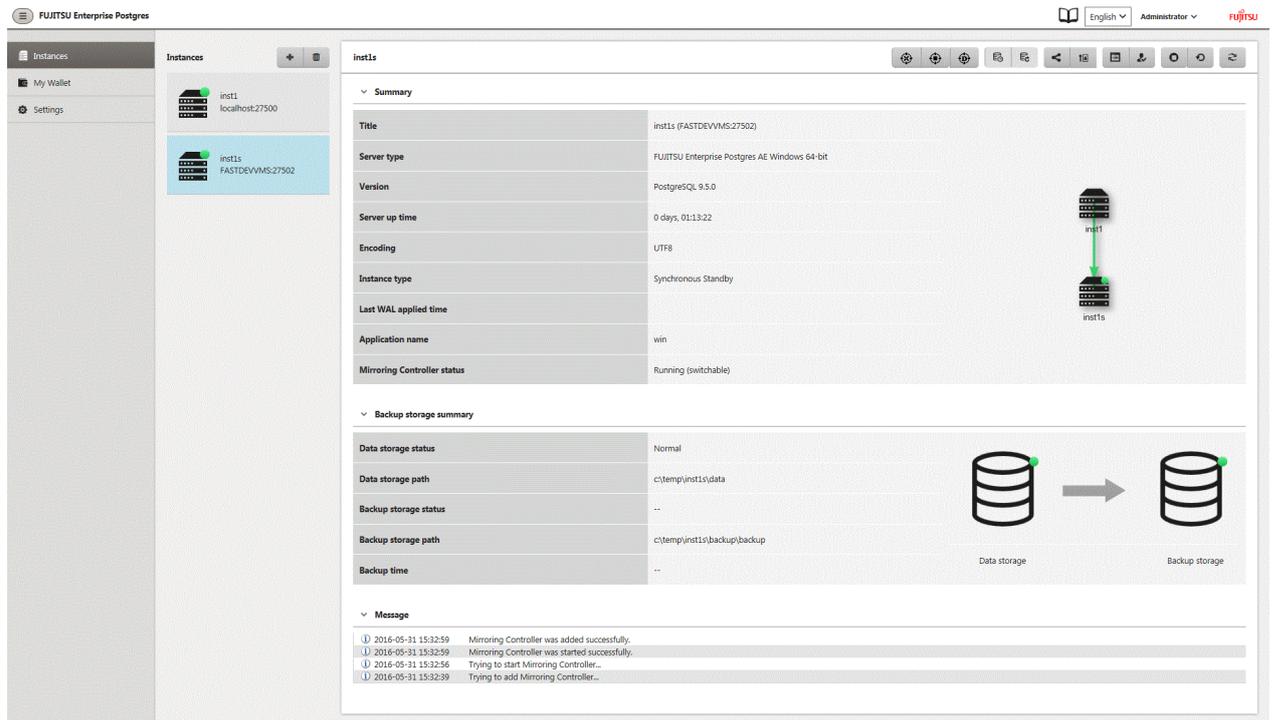
Enter the following items on master instance and on standby instance fields for Mirroring Controller setup, as shown in the above screenshot:

- [Enable automatic switch over]: Toggles the automatic switch/disconnection functionality. Select "Yes". The default is "No".
- [Mirroring Controller port]: Port number of Mirroring Controller.
- [Mirroring Controller management directory]: Directory where the Mirroring Controller configuration files will be stored.
- [Heartbeat interval (in milliseconds)]: Number of milliseconds between two consecutive heartbeat checks. The default is "800".
- [Heartbeat timeout (in 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".

4. Click  to setup Mirroring Controller.

5. Upon successful completion, Mirroring Controller will be started on master and standby instances.

In the [Instances] tab, select "inst1s". The following page will be displayed, with the Mirroring Controller status:



6.2 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.

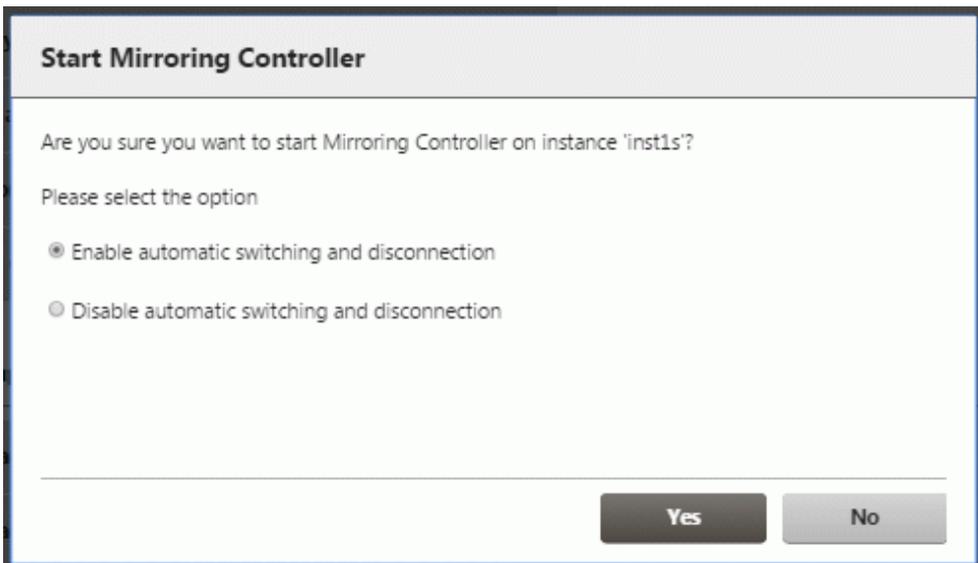
6.3 Starting Mirroring Controller

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.



4. In the confirmation dialog box, 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.

6.4 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.

6.5 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.

6.6 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].

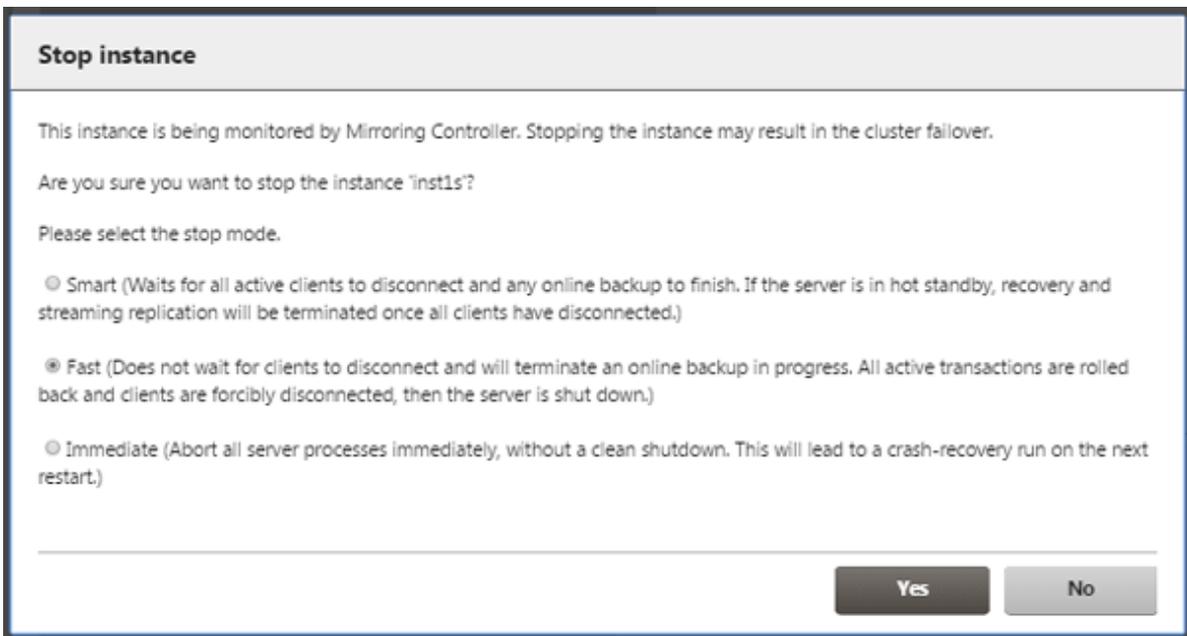
Mirroring Controller setup will be removed from the cluster. The cluster status will be updated and a confirmation message entry will be displayed in the [Message] section.

6.7 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 the 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.

6.8 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 refers to a situation where data update is performed on both servers because they both work as a primary server temporarily.

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 "[3.8.4 Tuning for Optimization of Degrading Operation Using Abnormality Monitoring](#)" 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 primary instance is selected in the [Instances] tab, and
4. "Standalone" is displayed in "Standalone".



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.

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	Literal "hot_standby"	Specify the output level for the transaction log. Specify "hot_standby".
max_wal_senders	Number of standby servers + 1	Specify the number of standby servers + 1. In FUJITSU Enterprise Postgres, 1:1 standby is used, so specify 2.
synchronous_standby_names	' <i>standbyServerName</i> '	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.
hot_standby	on	Specify whether queries can be run on the standby server. Specify this to execute reference jobs on the standby server. (*1)
wal_keep_segments	Number of file segments	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. Setting this parameter requires consideration regarding stabilization of the database multiplexing mode (refer to " 3.8.1 Tuning to Stabilize the Database Multiplexing Mode " for details). Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on estimating the WAL segment.
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 of $(\text{heartbeat_interval} + \text{heartbeat_timeout}/1000) \times (\text{heartbeat_retry} + 1)$ of the <i>serverIdentifier.conf</i> file, you

Parameter	Value set	Explanation
		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_xlogcopy.cmd" "%p" "backupDataStorageDestinationDir \archived_xlog\%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	The limit value for the number of standby server connections + number of simultaneous executions of mc_ctl status (*2) + 2 + 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.

*1: Mandatory to use the Mirroring Controller.

*2: 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
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.

Parameter	Value set	Explanation
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_command	'cmd /c ""installDir\bin\ \pgx_xlogcopy.cmd" "%p" "backupDataStorageDestinationD ir\archived_xlog\%f"'	Specify the command and storage destination to save the transaction log.
wal_receiver_timeout	Timeout (milliseconds)	Specify the time period after which it is determined that an error has occurred in the transaction log transfer on the standby server. By aligning this value with the value of (heartbeat_interval + heartbeat_timeout) x heartbeat_retry of the <i>serverIdentifier.conf</i> file, you can unify the time after which it is determined that an error has occurred.

A.3 Network Configuration File

Define the network configuration that will link the Mirroring Controller processes in the network configuration file (network.conf file).

Define the same content on the primary server and standby server.

Table A.3 network.conf file

Format specified	Explanation
<i>serverIdentifier hostName portNumber</i> <i>Or</i> <i>serverIdentifier ipAddress portNumber</i>	Specify any identifier (*1) for the server, a host name (or IP address), and port number (*2), using a space as the separator. Specify the IP address or corresponding host name (*3) that will be used to connect to the admin network. Define the same content on the primary server and standby server. Example) The literal <i><space></i> represents a space. [For IPv4] server1 <i><space></i> 192.0.2.100 <i><space></i> 27540 server2 <i><space></i> 192.0.2.110 <i><space></i> 27540 [For IPv6] server1 2001:258:8404:1217:250:56ff:fea7:559f 27000 server2 2001:258:8404:1217:250:56ff:fea7:55a0 27100

*1: The maximum length of the server identifier is 64 bytes. Use ASCII characters excluding spaces to specify this parameter.

*2: 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. Note that the value specified in this parameter must also be set in the services file.

*3: Use ASCII characters excluding spaces to specify the host name.

A.4 Server Configuration File

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	<i>'dataStorageDestinationDir'</i> [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.
db_instance_service_name	<i>'registeredServiceNameOfFujitsuEnterprisePostgresInstance'</i>	Specify the registered service name of the FUJITSU Enterprise Postgres instance in the Windows service. Use ASCII characters to specify this parameter.
db_instance_password	<i>'passwordOfInstanceAdminUser'</i>	Specify the password used when Mirroring Controller connects to a database instance. 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.
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. The value 0 indicates that there is no timeout limit. The default is 70000 milliseconds (70 seconds).
core_file_path	<i>coreFileOutputDir</i>	Specify the directory to which the core file is to be output. 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.
mc_service_name	<i>'registeredServiceNameOfMirroringController'</i>	Specify the Mirroring Controller service name registered in the Windows service. Use ASCII characters excluding forward slash (/) and backslash (\) to specify this parameter. The maximum length of the service name is 124 bytes. The default is 'MirroringControllerOpen'.
event_source(*1)	<i>'eventSourceName'</i>	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. The default is 'MirroringControllerOpen'.

Parameter	Value set	Explanation
heartbeat_interval	Interval time for operating system/server and process heartbeat monitoring, and disk abnormality monitoring (milliseconds)	Contact between servers, and abnormality monitoring, will be performed at the specified interval. Specify a value between 1 and 2147483647. The default is 800 milliseconds.
heartbeat_timeout	Timeout time for operating system/server and process heartbeat monitoring, and disk abnormality monitoring (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 1 second.
heartbeat_retry	Number of retries for operating system/server and process heartbeat monitoring, and disk abnormality monitoring (number of times)	Specify the number of retries 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 is 2 times.

*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.

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.

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 swith/disconnection

Start Mirroring Controller of the server "server1":

Example)

```
> mc_ctl start -M D:\mcdire\inst1 -w -f --local-server server1
```

Stop Mirroring Controller of the server "server1":

Example)

```
> mc_ctl stop -M D:\mcdire\inst1 --local-server server1
```

Disabling automatic swith/disconnection

Start Mirroring Controller of the server "server1":

Example)

```
> mc_ctl start -M D:\mcdire\inst1 -w --local-server server1
```

Stop Mirroring Controller of the server "server1":

Example)

```
> mc_ctl stop -M D:\mcdire\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 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.

C.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].

C.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) 2008:

1. In the [Start] menu, click [Control Panel].
2. Click [Security] and then click [Windows Firewall].
3. In the [Windows Firewall] window, click [Change settings].
4. On the [Exceptions] tab, click [Add port].
5. In the [Add a Port] window, set the following:
 - a. In [Name], specify any name.
 - b. In [Port number], specify the port number defined in the network definition file.
 - c. Select [TCP].
6. Click [OK].
7. On the [Exceptions] tab, in the [Program or port] list, check if the added port is enabled.

8. In the [Windows Firewall] window, click [OK].

Windows Server (R) 2008 R2:

1. In the [Start] menu, click [Control Panel].
2. Click [System and Security], and then click [Windows Firewall].
3. In the [Windows Firewall] window, click [Advanced settings] on the left side of the window.
4. In the [Windows Firewall with Advanced Security] window, click [Inbound Rules] on the left side of the window.
5. Click [New Rule] on the right side of the window.
6. In the [New Inbound Rule Wizard] window, select [Port], and then click [Next].
7. Select [TCP] and [Specific local ports], then specify the port number defined in the network definition file, and then click [Next].
8. Select [Allow the connection], and then click [Next].
9. Select the profiles for which this rule applies, and then click [Next].
10. In [Name], specify any name, and then click [Finish].
11. In the [Windows Firewall with Advanced Security] window, check if the added rule is enabled under [Inbound Rules] in the center of the window.

Windows Server(R) 2012 or Windows Server(R) 2012 R2:

1. Right-click the [Start] window, and then click [All apps] that is displayed at the bottom-right of the window. In the [Apps] window, click [Control Panel].
2. Click [Windows Firewall].
3. In the [Windows Firewall] window, click [Advanced settings] on the left side of the window.
4. In the [Windows Firewall with Advanced Security] window, click [Inbound Rules] on the left side of the window.
5. Click [New Rule] on the right side of the window.
6. In the [New Inbound Rule Wizard] window, select [Port], and then click [Next].
7. Select [TCP] and [Specific local ports], then specify the port number defined in the network definition file, and then click [Next].
8. Select [Allow the connection], and then click [Next].
9. Select the profiles for which this rule applies, and then click [Next].
10. In [Name], specify any name, and then click [Finish].
11. In the [Windows Firewall with Advanced Security] window, check if the added rule is enabled under [Inbound Rules] in the center of the window.

Index

[A]	[M]
About Database Multiplexing Mode..... 4	Manually Disconnecting the Standby Server..... 38
Action Required when All Servers or Instances Stopped.....60	Manually Switching the Primary Server..... 38
Action Required when an Error Occurs in the Database Multiplexing Mode..... 51	Matching the system times..... 13
Action Required when Automatic Disconnection Fails..... 60	Mirroring Controller Resources.....7
Action Required when Automatic Switch Fails..... 59	Monitoring Mirroring Controller Messages..... 39
Action Required when Server Degradation Occurs..... 51	Monitoring using database multiplexing mode..... 4
Addressing Errors During Degrading Operation..... 58	
Application Connection Server Settings.....28	[N]
Authentication of the Standby Server..... 11	network.conf file.....73
	Network Configuration File.....73
[B]	Notes on CPU Architecture and Products..... 9
Backing up Database Multiplexing Mode Information..... 33	Notes on Executing Reference Jobs on the Standby Server..... 9
Backup Data Storage Destination Directory..... 76	Notes on Indexes..... 28
Backup Operation..... 33	
	[O]
[C]	Operations in Database Multiplexing Mode.....35
Changes in Operation 45	Operations when the Server has Started Degrading after a Disconnection has Occurred.....56
Changes Required when the Standby Server is Stopped..... 45	Operations when the Server has Started Degrading after a Switch has Occurred..... 51
Changing from Database Multiplexing Mode to Single Server Mode..... 47	Overview of Database Multiplexing Mode..... 4
Changing from Single Server Mode to Database Multiplexing Mode..... 46	
Checking the Behavior..... 28	[P]
Checking the Database Multiplexing Mode Status..... 37	Parameters..... 71
Cluster Operation Provided by FUJITSU Enterprise Postgres...2	Parameters Set on the Primary Server..... 71
Configuration for Database Multiplexing Mode..... 6	Parameters Set on the Standby Server.....72
Confirming the Streaming Replication Status..... 27	postgresql.conf file..... 71,72
Creating, Setting, and Registering the Primary Server Instance19	Preparatory Tasks for the Output of Error Logs to the Event Log 14
Creating, Setting, and Registering the Standby Server Instance25	Preparing for Setup.....14
Creating Applications..... 27	Preparing the Backup Disk..... 14
[D]	[R]
Database Backup Operation..... 33	Rebuild the Standby Server..... 55,58
Definition of Cluster Operation..... 2	Recovering from an Incorrect User Operation..... 63
	Recovery of the Mirroring Controller management directory 54,58
[E]	Redundancy of the Admin and Log Transfer Networks.....8
Encryption of Transaction Logs Transferred to the Standby Server 12	Registering Service Names and Event Source Names in the Windows Service..... 76
	Rolling Updates..... 40
[F]	
Failback of the Primary Server..... 55	[S]
	Scalability..... 2
[H]	Security in Database Multiplexing..... 9
High availability..... 2	Security Policy Settings..... 15,78
How to Execute the mc_ctl Command..... 76	ServerConfiguration File..... 74
	serverIdentifier.conf file..... 74
[I]	Server Maintenance..... 40
Identify cause of error and perform recovery..... 54,58	Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances..... 32
Identify Cause of Error and Restore the Standby Server..... 53	Setting up Database Multiplexing Mode..... 13
Identify Cause of Error and Restructure the Standby Server... 57	Setting up Database Multiplexing Mode on the Primary Server15
Installation..... 14	Setting up Database Multiplexing Mode on the Standby Server22
	Setting up the Primary Server..... 15
[L]	Setting up the Standby Server..... 22
Log shipping mode..... 2	

Setup.....	13
Starting and Stopping Instances.....	35
Starting Mirroring Controller.....	35
Stop Mirroring Controller.....	53,57
Stopping for Maintenance	45
Stopping Mirroring Controller.....	36
Supplementary Information on Building the Primary Server and Standby Server on the Same Server.....	76
Supplementary Procedure on Configuring for Operation in Database Multiplexing Mode.....	78

[T]

Tuning.....	28
Tuning for Optimization of Degrading Operation Using Abnormality Monitoring.....	29
Tuning to Stabilize Queries on the Standby Server.....	28
Tuning to Stabilize Queries on the Standby Server (when Performing Frequent Updates on the Primary Server).....	29
Tuning to Stabilize the Database Multiplexing Mode.....	28

[U]

Uninstalling in Database Multiplexing Mode.....	49
Users who perform setup and operations.....	13

[W]

What is a cluster system?	2
Windows Firewall Settings.....	78