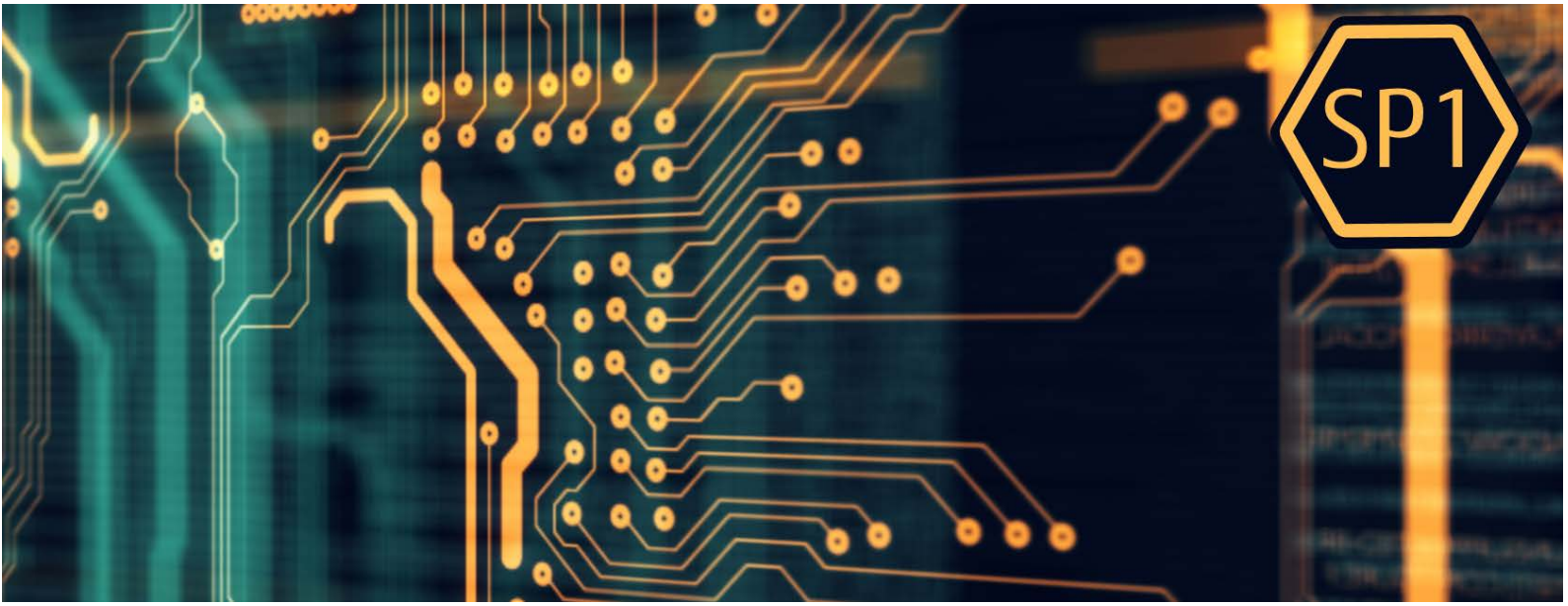
FUJITSU

FUJITSU Enterprise Postgres 13



Reference Guide

Windows/Linux

J2UL-2683-02ENZ0(00)
August 2021

Preface

Purpose of this document

This document is a command reference, and explains FUJITSU Enterprise Postgres commands and options with features expanded on from PostgreSQL.

Intended readers

This document is aimed at people who manage and operate FUJITSU Enterprise Postgres. Readers of this document are also assumed to have general knowledge of:

- PostgreSQL
- SQL
- Linux
- Windows

Structure of this document

This document is structured as follows:

Chapter 1 Command List and Specification Format

Lists commands and describes their specification format.

Chapter 2 Client Commands

Explains options not listed in "PostgreSQL Client Applications" in the PostgreSQL Documentation.

Chapter 3 Server Commands

Explains commands and options not listed in "PostgreSQL Server Applications" in the PostgreSQL Documentation.

Chapter 4 Mirroring Controller Commands

Explains the Mirroring Controller commands.

Chapter 5 Connection Manager Commands

Explains the Connection Manager commands.

How to read this document

Examples in this document are predominantly for UNIX/Linux.

For Windows, replace values (such as paths in the examples) as appropriate.

Export restrictions

If this document is to be exported or provided overseas, confirm legal requirements for the Foreign Exchange and Foreign Trade Act as well as other laws and regulations, including U.S. Export Administration Regulations, and follow the required procedures.

Issue date and version

```
Edition 2.0: August 2021
Edition 1.0: April 2021
```

Copyright

Copyright 2015-2021 FUJITSU LIMITED

Contents

Chapter 1 Command List and Specification Format....1
1.1 Command List....1
1.1.1 Client Commands....1
1.1.2 Server Commands....1
1.1.3 Mirroring Controller Commands....1
1.1.4 Connection Manager Commands....2
1.2 Command Specification Format....2

Chapter 2 Client Commands....3
2.1 pg_dumpall....3
2.2 pgx_loader....3

Chapter 3 Server Commands....6
3.1 pg_ctl....6
3.2 pgx_dmpall....6
3.3 pgx_keystore....8
3.4 pgx_rcvall....9
3.5 postgres....12

Chapter 4 Mirroring Controller Commands....13
4.1 mc_arb....13
4.2 mc_ctl....15

Chapter 5 Connection Manager Commands....20
5.1 cm_ctl....20

Chapter 1 Command List and Specification Format

This chapter lists commands and describes their specification format.

1.1 Command List

This chapter lists commands and options not explained in "PostgreSQL Client Applications" or in "PostgreSQL Server Applications" in the PostgreSQL Documentation.

1.1.1 Client Commands

The commands below have options not explained in "PostgreSQL Client Applications" in the PostgreSQL Documentation.

Command	Functional overview
pg_dumpall	Extract a PostgreSQL database cluster into a script file

The command below is not explained in "Client Applications" in the PostgreSQL Documentation.

Command	Functional overview
pgx_loader	Loads data from an external file into a PostgreSQL table.

1.1.2 Server Commands

The commands below have options not explained in "PostgreSQL Server Applications" in the PostgreSQL Documentation.

Command	Functional overview
pg_ctl	Initialize, start, stop, or control a PostgreSQL server
postgres	PostgreSQL database server

The commands below are not explained in "PostgreSQL Server Applications" in the PostgreSQL Documentation.

Command	Functional overview
pgx_dmpall	Backs up the data directory, tablespaces, and configuration files.
pgx_keystore	Manages keystore
pgx_rcvall	Recovers the data directory, tablespaces, and configuration files.

1.1.3 Mirroring Controller Commands

Mirroring Controller has the following commands:

Command	Functional overview
mc_arb	L Start, stop and display the status of the Mirroring Controller arbitration process. W Start, stop and display the status of the Mirroring Controller arbitration process, or register and unregister as a Windows service.

Command	Functional overview
mc_ctl	L Start and stop Mirroring Controller, switch/ disconnect the server, or display the server status. W Start and stop Mirroring Controller, switch/ disconnect the server, display the server status, or register and unregister as a Windows service.

1.1.4 Connection Manager Commands

Connection Manager has the following commands:

Command	Functional overview
cm_ctl	Start, stop or display the status of the conmgr process

1.2 Command Specification Format

The table below shows the command specification format.

Item	Explanation
[ ]	Indicates optional element.
...	Indicates that the item can be specified repeatedly.

Chapter 2 Client Commands

This chapter explains options not listed in "PostgreSQL Client Applications" in the PostgreSQL Documentation.

2.1 pg_dumpall

Name

pg_dumpall -- Extract a PostgreSQL database cluster into a script file

Synopsis

```
pg_dumpall [connectionOption...] [option...]
```

Options

--no-tablespace-encryption

Do not output commands to encrypt tablespaces. Running the generated SQL script will restore the originally encrypted data without being encrypted.

See

Refer to "pg_dumpall" in the PostgreSQL Documentation for details.

2.2 pgx_loader

Name

pgx_loader --Loads data from a file into a PostgreSQL table.

Overview

```
pgx_loader load -c command [options...]
```

```
pgx_loader recovery -t table
```

Description

The pgx_loader command loads data from an external file into PostgreSQL tables, and commits or rolls back transactions prepared during data load.

In load mode, data is loaded at high speed by executing the COPY FROM command specified in *command* at a certain degree of parallelism. If load is completed successfully, the message below is output to the standard output.

```
LOAD count
Note: count is the number of rows loaded.
```

In recovery mode, commit or rollback of transactions prepared during data load is performed.

Options

-a
--echo-sql

Display the executed command in the standard output.

-c *command*
--copy-command=*command*

Specify the COPY FROM command to be executed. If STDIN is specified for the FROM clause, data will be loaded from the standard input. In this case, SQL superuser privileges (or having one of the roles pg_read_server_files or

pg_execute_server_program) are not required, because local user access privileges will be used for external files and external programs, instead of database server access privileges.

"binary" cannot be specified for the FORMAT option of the COPY FROM command specified in this option. Therefore, input files in binary format cannot be specified.

The FREEZE option cannot be specified for the COPY FROM command specified in this option.

See

Refer to "COPY" in the PostgreSQL Documentation for information on the COPY FROM command.

-j *number-of-jobs*
--jobs=*number-of-jobs*

Specify the number of background workers (parallel workers) that the COPY COMMAND should use to simultaneously perform data conversion, table creation, and index creation. This option can dramatically reduce the time for loading a large amount of data on instances that runs on multiple processor machines.

The optimal value depends on the server, client, and network configurations. The number of CPU cores and disk configuration also affect the optimal value. The number of CPU cores of the server is recommended as the initial value to try. Naturally, if a value that is too large is used, performance degradation will occur due to thrashing and context switching.

Specify a value from 2 to 128. The default is 2.

-t *table*
--table=*table*

Complete the prepared transactions only for the specified table.

-?
--help

Show how to use pgx_loader command line arguments, and exit.

The command line options below control the database connection parameters.

-d *connstr*
--dbname=*connstr*

Specify the database name to connect to.

If this option is not specified, the PGDATABASE environment variable will be used. If the environment variable is not set, your operating-system user name will be used.

-h *host*
--host=*host*

Specify the host name of the machine the database server runs on. If the specified value starts with a slash, it will be used as the directory for a Unix domain socket.

If this option is not specified, the PGHOST environment variable will be used. If the environment variable is not set, it will be considered a Unix domain socket connection.

-p *port*
--port=*port*

Specify the TCP port to be used by the server to monitor the connection, or extension of the local Unix domain socket file.

If this option is not specified, the PGPORT environment variable will be use. If the environment variables is not set, 27500 will be used.

-U *username*
--username=*username*

User name for connection to the database.

-w
--no-password

Never prompt for the password. If the server requires password authentication but other means (such as a .pgpass file) are not available, the connection attempt will fail. This option can be useful in batch jobs, scripts, and so on, where no user is present to enter a password.

-W
--password

Force pgx_loader to prompt for the password before connecting to the database. This option is never essential, since pgx_loader will automatically prompt for the password if the server demands password authentication. However, pgx_loader will waste a connection attempt finding out if the server requires a password. In some cases it is worth specifying this option to avoid the extra connection attempt.

Diagnostics

load mode

0: Normal exit

Other than 0: Abnormal exit

recovery mode

0: There are no prepared transactions that must be completed

3: A prepared transaction was committed

4: A prepared transaction was rolled back

Other than the above: Abnormal exit

The order of the table rows loaded by pgx_loader may not match the order of the file lines. This is because the file lines will have been inserted into the table in parallel, by multiple parallel workers.

Example

The example below loads the file /path/to/data.csv (2000 records) into table tbl using a degree of parallelism of 3.

```
$ pgx_loader load -j 3 -c "COPY tbl FROM '/path/to/data.csv' WITH CSV"
LOAD 2000
```

The example below reads the file /path/to/data.csv (2000 records) from the standard input and loads into table tbl using a degree of parallelism of 3.

```
$ pgx_loader load -j 3 -c "COPY tbl FROM STDIN WITH CSV" < /path/to/data.csv
LOAD 2000
```

The example below completes the transactions prepared for table tbl.

```
$ pgx_loader recovery -t tbl
```

Chapter 3 Server Commands

This chapter explains commands and options not listed in "PostgreSQL Server Applications" in the PostgreSQL Documentation.

3.1 pg_ctl

Name

pg_ctl -- Initialize, start, stop, or control a PostgreSQL server

Synopsis

```
pg_ctl start [-D datadir] [-l filename] [-W] [-t seconds] [-s]
           [-o options] [-p path] [-c] [--keystore-passphrase]

pg_ctl restart [-D datadir] [-m s[mart] | f[ast] | i[mmediate] ]
             [-W] [-t seconds] [-s] [-o options] [-c]
             [--keystore-passphrase]
```

Options

--keystore-passphrase

Prompt for the passphrase to open the keystore.

See

Refer to "pg_ctl" in the PostgreSQL Documentation for details.

3.2 pgx_dmpall

Name

pgx_dmpall - Backs up the data directory, tablespaces, and configuration files.

Synopsis

```
pgx_dmpall  [option...]
```

Description

The pgx_dmpall command backs up the data directory, tablespaces, and configuration files. The backup data is stored in the directory specified by backup_destination parameter of postgresql.conf. The pgx_dmpall command also deletes archived Write Ahead Logs (WAL) that are no longer necessary for recovery when the backup completes successfully.

Options

-c

This option only backs up configuration files. The configuration files are as follows:

- postgresql.conf (postgresql.conf)
- File for host-based authentication (pg_hba.conf)
- Configuration file for ident authentication (pg_ident.conf)

If an external reference, such as 'include' in postgresql.conf, is set, the reference destination files are also backed up.

-C fast|spread
--checkpoint=fast|spread

Sets checkpoint mode to fast or spread (default).
If fast is specified, the checkpoint processing at the start of backup becomes quick, but the impact on performance of running applications gets larger due to intense I/O. In spread mode, the impact on applications is smaller but the backup takes longer, because the checkpoint is performed slowly.

-D *datadir*

Specify the data directory. If this option is omitted, the value of the environment variable PGDATA is used.

-f *configFile*

Specify the postgresql.conf configuration file. This option is set if the data directory and the configuration file set in the 'data_directory' parameter of the postgresql.conf file are running in separate directories.

-P *tablespacesListFile*
--tablespaces-list-file=*tablespacesListFile*

Specify the full path of the file containing the names of the tablespaces to be backed up using the copy command, using less than 1024 bytes.

The file format is described below:

tablespaceName<newline>
tablespaceName<newline>
...

Tablespaces not listed in the specified file are backed up to the backup storage directory. If this option is not specified, all tablespaces are backed up using the copy command.
This option can be specified if the -Y option has been specified, and it is used to limit the tablespaces backed up using the copy command.

-U *username*
--username=*username*

Specify the user name of the database superuser. This defaults to the name of the effective user running pgx_dmpall.

-Y *copyCommandFile*
--copy-command=*copyCommandFile*

Specify the full path of the file of the copy command for backup, using less than 1024 bytes. This option cannot be specified together with the -c option.

-w
--no-password

Never issue a password prompt. If the server requires password authentication and a password is not available by other means such as a .pgpass file, the connection attempt will fail. This option can be useful in batch jobs and scripts where no user is present to enter a password.

-W
--password

Force pgx_dmpall to prompt for a password before connecting to a database.

This option is never essential, since pgx_dmpall will automatically prompt for a password if the server demands password authentication. However, pgx_dmpall will waste a connection attempt finding out that the server wants a password. In some cases it is worth typing -W to avoid the extra connection attempt.

--maintenance-db=*dbname*

Specifies the name of the database to connect to. If not specified, the postgres database will be used; if that does not exist, template1 will be used.

Any database can be specified as long as it can be connected to.

--exclude-copy-cluster

Excludes a database cluster from backup via the copy command. This option can be specified if the -Y option has been specified. If this option is not specified, the database cluster will be backed up using the copy command.

Environment

PGDATA

Specify the data directory. You can overwrite using the -D option.

Diagnostics

0: Normal end

Other than 0: Abnormal end

Notes

This command can only be executed when the database server is running.

Execute this command as a PostgreSQL user account.

Do not update or delete files in the backup storage directory. Otherwise, you may not be able to recover the database.

Do not store other files in the backup storage directory.

This command uses one database connection. To establish a connection, this command uses the IPv4 loopback address 127.0.0.1 on Windows, and the UNIX domain socket on other operating systems. Therefore, permit these connections in pg_hba.conf.

This command cannot be executed on the standby server.

Example

In the following example, the data directory, tablespaces, and configuration files are backed up. At this time, stored WALs are no longer necessary because the backups are destroyed.

```
$ pgx_dmpall
```

Related item

pgx_rcvall

3.3 pgx_keystore

Name

pgx_keystore -- Manages keystore

Synopsis

```
pgx_keystore [option...] keystore_location
```

Description

pgx_keystore enables auto-open of a keystore.

Options

-a
--enable-auto-open

Enables auto-open of a keystore. This allows the keystore to open automatically without entering the passphrase when the database server starts.

When auto-open is enabled, an obfuscated copy keystore.aks is created in the same directory where the keystore file keystore.ks is stored. To disable auto-open, delete keystore.aks.

-P *passphrase*
--passphrase=*passphrase*

Specify the passphrase to open the keystore. If this option is omitted, the prompt to enter the passphrase is displayed.

keystore_location

Specify the absolute or relative path of the keystore file.

Diagnostics

0: Normal exit

Other than 0: Abnormal exit

Notes

This command can be executed whether the database server is running or stopped. It does not connect to the database server.

Example

Enables auto-open of a keystore.

```
$ pgx_keystore -a /key/store/location/keystore.ks
```

3.4 pgx_rcvall

Name

pgx_rcvall - Recovers the data directory, tablespaces, and configuration files.

Synopsis

```
pgx_rcvall [option...]
```

Description

The pgx_rcvall command recovers the data directory, tablespaces, and configuration files using the data that was backed up with pgx_dmpall command and archived Write-Ahead-Log (WAL). If none of the options that indicate the recovery point is specified, all archived WAL are applied and the data will be recovered to the latest point.

Options

-B *backupdir*

Specify the backup storage directory. If the data directory is damaged, this option cannot be omitted.

-D *datadir*

Specify the data directory. If this option is omitted, the value of the environment variable PGDATA is used.

-e *targetTime*

Specify this option to recover the data as of the specified date and time.

targetTime

Specify the time at which the data is recovered. The format is as follows:

```
"YYYY-MM-DD HH:MM:SS"
```

-l

This option displays a list of the backup data information in the backup storage directory that was obtained using the pgx_dmpall command. If the pgx_dmpall command was executed using the copy command (-Y option) for backup, the resources backed up using the copy command will also be listed. This cannot be specified together with -p, -e or -n option.

-n *restorePoint*

Specify this option to recover the data to the specified restore point. Restore points are created with SQL function pg_create_restore_point. If multiple restore points with the same names were created, the first one after the backup was taken is used for recovery. If the specified restore point does not exist, the recovery fails. This cannot be specified together with -e or -p option.

-p

Specify this option to recover the data as of the time when the last backup completed. This cannot be specified together with -e or -n option.

-x

Specify this option if you do not want to include transactions committed at the time specified in the -e option as part of the recovery.

-Y *copyCommandFile*
--copy-command=*copyCommandFile*

Specify the full path of the file of the copy command for recovery, using less than 1024 bytes. This option cannot be specified together with the -l option.

--keystore-passphrase

Prompt for the passphrase to open the keystore.

--view-results-of-copying

Output the backup information file that was written by the copy command executed via the pgx_dmpall command. This option cannot be specified together with the -l, -p, -e, -n, or -Y option.

Environment

PGDATA

Specify the data directory. You can overwrite using the -D option.

PGPORT

Specify the port number for connecting to the database.

PGUSER

Specify the user name of the database superuser. This defaults to the name of the effective user running pgx_dmpall.

Diagnosis

0: Normal exit

Other than 0: Abnormal exit

Backup data information

Date

Date the backup data was created using the pgx_dmpall command.

Dir

This is the name of the directory in the backup storage directory that is used to store the backup data.
Directory naming format: Time format (YYYY-MM-DD_HH-MM-SS)

Status

This is the status of the pgx_dmpall command backup data.

COMPLETE: Complete

INCOMPLETE: Incomplete

Resources backed up by the copy command

List of resources that were backed up by the copy command executed via the pgx_dmpall command.

If there are resources that were backed up by the copy command, then database clusters ('pg_data') or tablespace names will be listed, delimited by header and halfwidth comma.

Notes

This command can only be executed when the database server is stopped, except when it is executed with -l option.

Execute this command as a PostgreSQL user account.

Use backup data that was taken from the recovery target data directory.

Before executing this command, disconnect all application database connections. Additionally, do not connect to the database during recovery.

The configuration files are restored from those files that were taken by the last pgx_dmpall (including -c option).

This command connects to the database to determine whether the recovery has completed. So ensure that you set the port number with PGPORT environment variable in the environment where multiple instances exist.

Match the OS timezone setting when running pgx_dmpall/pgx_rcvall to the timezone specified by timezone parameter in postgresql.conf.

Otherwise, data might be recovered to an unexpected time when -e or -p is specified.

If you recover to a past point, a new timeline (history of database updates) begins at that point. That recovery point is the latest point in the new timeline when the recovery is completed. If you subsequently recover to the latest point, the database updates in the new timeline will be replayed.

Valid restore points are the ones that were created in the timeline where the backup had been taken. That means that if you recover to a past point, those restore points created thereafter are unavailable. Therefore, take a backup when you have restored the past data desired.

If the pgx_dmpall command was executed using the copy command (-Y option) for backup, it is necessary to execute this command using the copy command (-Y option) for recovery. However, because the list of resources (database cluster or tablespace) that were backed up using the copy command is recorded in the backup directory, there is no need to specify the target resources when executing this command. The -I option can be used to check the target resources for which a backup is retrieved using the copy command.

Example

In the following example, the data directory, tablespaces, and configuration files are recovered.

```
$ pgx_rcvall -B /home/pgsql/Backupdir
```

In the following example, the data directory and tablespaces are recovered at 10:00:00 on 01-05-2017. The configuration files are recovered at the point at which the last of the data is obtained.

```
$ pgx_rcvall -B /home/pgsql/Backupdir -e "2017-05-01 10:00:00"
```

In the following example, the data directory and tablespaces are recovered up to the time of restore point "before_match_20170510_1". The configuration files are restored from the latest backup.

```
$ pgx_rcvall -B /home/pgsql/Backupdir -n before_match_20170510_1
```

In the following example, the obtained backup data information in the backup storage directory is displayed in a list.

```
$ pgx_rcvall -l
```

Related item

pgx_dmpall

3.5 postgres

Name

postgres -- PostgreSQL database server

Synopsis

```
postgres [option...]
```

Options

-K

Prompt for the passphrase to open the keystore. This option works in single-user mode only, so you must also specify the --single option, as shown below:

postgres --single -K

See

Refer to "postgres" in the PostgreSQL Documentation for details.

Chapter 4 Mirroring Controller Commands

This chapter explains the Mirroring Controller commands.

4.1 mc_arb

Name

L mc_arb - Start, stop, and display the status of the Mirroring Controller arbitration process

W mc_arb - Start, stop, and display the status of the Mirroring Controller arbitration process, or register and unregister it as a Windows service

Overview

```
mc_arb start [-M mcdir] [-w| -W]

mc_arb stop [-M mcdir] [-e]

mc_arb status [-M mcdir]
```

W
```
mc_arb register [-M mcdir] [-w| -W] -P password [-S a[uto] | d[emand]]
mc_arb unregister [-M mcdir]
```

Description

L mc_arb starts, stops, and displays the status of the Mirroring Controller arbitration process.

W mc_arb starts, stops, and displays the status of the Mirroring Controller arbitration process, or registers and unregisters it as a Windows service.

The start mode starts the Mirroring Controller arbitration process.

The stop mode stops the Mirroring Controller arbitration process.

The status mode displays the connection status of the Mirroring Controller arbitration process with the Mirroring Controller processes of the primary server and standby server.

W The register mode registers the Mirroring Controller arbitration process as a Windows service. The -w and -W options can be used to start/stop the Mirroring Controller arbitration process through the Windows service, and when it executes the mc_arb command, the specified option will take effect.

The unregister mode unregisters the Mirroring Controller arbitration process as a Windows service.

If the Mirroring Controller arbitration process has not been started on the server executing the command, stop mode and status mode will terminate with an error.

Additionally, if Mirroring Controller is forcibly stopped on the database server, it may take a few moments until the status mode displays the status of the server connection as offline.

L This command can be executed by any user.

W Execute this command as a user with administrator privileges (operating system user that belongs to the "Administrators" group).

Options

-e

Specify this option to forcibly stop the Mirroring Controller arbitration process on the active server.

Specify this option to stop the Mirroring Controller arbitration process but keep Mirroring Controller running (to stop both, first stop Mirroring Controller of the database server, and then the Mirroring Controller arbitration process). This option can also be specified to halt the arbitration process (by stopping the Mirroring Controller arbitration process) when the fencing command called by the arbitration process becomes unresponsive.

-M *mcdir*

Specify the Mirroring Controller arbitration process management directory.

ASCII characters can be specified in the directory path.

If this option is omitted, the value of the environment variable ARBCONTROLDIR is used.

W

-P *password*

For the register mode, specify the password for the user who executed the command.

W

-S a[uto] | d[emand]

Specify the start type for the Windows service to be registered. You can choose auto or demand as the start type by specifying the entire word or just its first letter. The default is auto.

-w

Waits for operations to finish.

L

This option is the default of start mode.

W

This option is the default of start mode and register mode.

-W

Does not wait for operations to finish.

Environment

ARBCONTROLDIR

Specify the Mirroring Controller arbitration process management directory.

ASCII characters can be specified in the directory path.

You can specify the -M option to override this value.

Diagnostics

0: Normal end

Other than 0: Abnormal end

Notes

If the Mirroring Controller arbitration process is forcibly stopped or communication between the command and the Mirroring Controller arbitration process is interrupted while the Mirroring Controller arbitration process is being stopped, a message that the command is being executed may be output and stopping may terminate in error, even though no other instances of the mc_arb command are being executed.
To solve this issue, ensure that other instances of the mc_arb command are not being executed before forcibly stopping the Mirroring Controller arbitration process.

W

Use the start mode and stop mode to start and stop the Windows service. Before being able to use these modes, use the register mode to register the Mirroring Controller arbitration process as a Windows service.
The mc_arb command must be executed from "Administrator: Command Prompt". Right-click [Command Prompt], and then select [Run as administrator] from the context menu to display the [Administrator: Command Prompt] window.

Example

L

Start the Mirroring Controller arbitration process.

```
$ mc_arb start -M /mcarb_dir/arbiter1
```

Start the Mirroring Controller arbitration process.

```
> mc_arb start -M D:\mcarb_dir\arbiter1
```

Display details of mc_arb status

```
server_id   host            status
(1)         (2)             (3)
```

```
(1) Server identifier
(2) Host name or IP address
(3) Server connection status
    online   : Connected
    offline  : Disconnected
```

4.2 mc_ctl

Name

mc_ctl - Start and stop Mirroring Controller, switch/disconnect the server, or display the server status.

mc_ctl - Start and stop Mirroring Controller, switch/disconnect the server, display the server status, or register and unregister as a Windows service.

Overview

```
mc_ctl start [-M mcdir] [-w| -W] [-f| -F] [--mc-only] [--async-connect-arbiter] [--local-server server_id]

mc_ctl stop [-M mcdir] [[-a] [--mc-only]| [-e][--local-server server_id]]

mc_ctl status [-M mcdir] [--arbiter] [--local-server server_id]

mc_ctl switch [-M mcdir] [--force [--no-fencing ]] [--local-server server_id]

mc_ctl detach [-M mcdir] [--no-fencing] [--local-server server_id]

mc_ctl enable-failover [-M mcdir] [--local-server server_id]

mc_ctl disable-failover [-M mcdir] [--local-server server_id]
```

```
mc_ctl register [-M mcdir] [-w| -W] [-f| -F] [--mc-only] -P password
                [-S a[uto] | d[emand]] [--async-connect-arbiter] [--local-server server_id]
```

```
mc_ctl unregister [-M mcdir] [--local-server server_id]
```

Description

mc_ctl starts and stops Mirroring Controller, switches/disconnects the server, or displays the server status.

mc_ctl starts and stops Mirroring Controller, switches/disconnects the server, displays the server status, or registers and unregisters as a Windows service.

The start mode starts Mirroring Controller. If the --mc-only option is omitted, the command starts a database instance.

The stop mode stops Mirroring Controller. If the --mc-only option is omitted, the database instance is stopped. If --mc-only option is not specified, database instance is also stopped. When executes on standby server without --mc-only, standby server will be detached from primary server.

The status mode displays the status of the servers, database instance processes, and disks monitored by Mirroring Controller. Additionally, if the --arbiter option is specified, Mirroring Controller arbitration process connection status is displayed.

The switch mode switches the primary server. When the server is switched, the database instance on the primary server stops, and the database instance on the standby server is upgraded to primary server and begins degrading operation. This mode can be executed on the primary server and standby server in an environment where the Mirroring Controller process can communicate with the Mirroring Controller process on the other server.

The detach mode forcibly disconnects the standby server. This mode is used to forcibly disconnect the other server when stopping of Mirroring Controller cannot be performed using stop mode (which requires login to the standby server). It can only be executed on the primary server.

The enable-failover mode enables automatic switching and disconnection.

The disable-failover mode disables automatic switching and disconnection.

W

The register mode registers Mirroring Controller as a Windows service. The -w, -W, -f, -F, --mc-only, --async-connect-arbiter, and --local-server options are used when Mirroring Controller is started and stopped as a Windows service. If the mc_ctl command is used to start and stop Mirroring Controller, the option specified in the command will be valid.

The unregister mode unregisters Mirroring Controller as a Windows service.

L

If Mirroring Controller has not been started on the server that executes the command, commands for any mode other than the start mode, and status mode.

Execute this command as an instance administrator.

W

If Mirroring Controller has not been started on the server that executes the command, commands for any mode other than the start mode, status mode, register mode, and unregister mode terminate with an error.

Execute this command as a user with the "Administrator" privilege (operating system user ID that belongs to the Administrator group).

Until you start Mirroring Controller of standby server after starting Mirroring Controller of the primary server, operation can be started with only the primary server. Standby server is incorporated when you start the Mirroring Controller of standby server, and you should be able to operate in the multiplexing configuration.

Options

-a

Specify this option to stop Mirroring Controller on all servers.

-e

Specify this option to forcibly stop Mirroring Controller on the active server.

-f

Specify this option to enable automatic switching and disconnection of Mirroring Controller immediately after startup.

L

This option is the default of start mode.

W

This option is the default of start mode and register mode.

-F

Specify this option to disable automatic switching and disconnection immediately after startup of Mirroring Controller.

--async-connect-arbiter

Specify this option to connect the Mirroring Controller start process asynchronously to the Mirroring Controller arbitration process. This option can be specified to forcibly start Mirroring Controller if the Mirroring Controller arbitration process is not started.

Specify this option if using the Mirroring Controller arbitration server.

--arbiter

Specify this option to display the connection status of the Mirroring Controller arbitration process. This option can be specified if using status mode.

Specify this option if using the Mirroring Controller arbitration server.

--local-server *server_id*

If you run a simulation build of the primary and standby servers in a single server (for system testing, for example), specify this option to identify the server to be operated.

For *server_id*, specify the server identifier specified in the network.conf file. ASCII characters other than single-byte space can be specified in the server identifier. The operations will be executed as if the user has logged in to *server_id*.

--mc-only

Specify this option to start and stop only Mirroring Controller processes. At the start mode, this option can be specified only while the database instance is running. If this option is omitted, the database instance is simultaneously started and stopped.

-M *mcdir*

Specify the Mirroring Controller management directory.

ASCII characters can be specified in the directory path.

If this option is omitted, the value of the environment variable MCCONTROLDIR is used.

--force

Switching with this option specified can only be specified on the standby server. This option is used to perform switching forcibly after performing fencing on the primary server if communication with the Mirroring Controller process of the other server is not possible (due to a network issue between database servers or unresponsive server, for example), thus preventing normal switching. However, if the --no-fencing option is specified, fencing will not be performed on the primary server.

--no-fencing

When switching or disconnection is executed with the --force option specified, fencing of the server to be degraded is circumvented. Therefore, it is necessary for the user to isolate the server to be degraded from the cluster system in advance.

-P *password*

For the register mode, specify the password for the user who executed the command.

-S a[uto] | d[emand]

Specify the start type for the Windows service to be registered. You can choose auto or demand as the start type by specifying the entire word or just its first letter. The default is auto.

-w

Waits for operations to finish.

This option is the default of start mode.

This option is the default of start mode and register mode.

-W

Does not wait for operations to finish.

Environment variable

MCCONTROLDIR

Specifies the Mirroring Controller management directory.

ASCII characters can be specified in the directory path.

You can specify the -M option to override this value.

Diagnostics

0: Normal end

Other: Abnormal end

Notes

The message under execution might be output though the mc_ctl command is not being executed and, besides, it terminate abnormally when the server is downed while processing execution of this command, an automatic switch, and an automatic separation, and the communication between a primary server and the standby server is cut off. Besides, restart Mirroring Controller to solve this problem after confirming that the mc_ctl command is not in progress. Afterwards, execute a necessary operation.

If a time-out error occurs when the mc_ctl command is in progress, the messages may be different from the processes. Take the actions described in the "Action" section of the message.

Automatic switching and disconnection by the enable-failover mode, the disable-failover mode, and the start mode can be enabled/disabled only while Mirroring Controller is running. Therefore, if you want to enable/disable automatic switching and disconnection on starting, specify the -f option or -F option each time you start Mirroring Controller.

W

Use the start mode and stop mode to start and stop the Windows service. To do this, use the register mode to register Mirroring Controller as a Windows service in advance.

In case of postgresql.conf has any incorrect parameter when this command is executed, this command may be abnormally terminated. If this is the case, re-execute it again after correct the parameter in postgresql.conf.

If the arbitration server of Mirroring Controller is used, connection with the arbitration server will be performed on startup even if startup using start mode is executed with the -F option specified, or if executed with --local-server specified.

W

The mc_ctl command must be executed from "Administrator: Command Prompt". Right-click [Command Prompt], and then select [Run as administrator] from the menu to display the [Administrator: Command Prompt] window.

For operation using an arbitration server, the amount of time spent attempting to connect with the arbitration server is calculated, so the Mirroring Controller startup time may take longer.

Example

L

To start Mirroring Controller:

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

W

To start Mirroring Controller:

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

Display details of mc_ctl status

```
mirroring status
----------------
(1)

server_id  host_role        host          host_status  db_proc_status       disk_status
(2)        (3)               (4)             (5)          (6)                    (7)
```

```
(1) Multiplexing status
    switchable        : Switchable
    switching         : Switching
    switched        : Switched (displayed when switching has finished and the degrading
operations status has been enabled)
    not-switchable    : Not switchable (displayed when a server is disconnected and
switching is not possible)
    unknown           : Unknown (*1)
    failover-disabled : Failover is disabled
(2) Server identifier
(3) Server role
    primary : Primary
    standby : Standby
    none(inactivated primary): No role
                              (primary is stopped or being defined as primary)
    none(inactivated standby): No role
                               (standby is stopped or being defined as primary)
(4) Host name or IP address
(5) Live/dead state of the server
    normal   : Normal operation
    abnormal : Abnormal
    unknown  : Unknown (*1)
(6) DBMS process status
    normal                         : Normal
    abnormal (abnormal process name (*2)) : Abnormal
    unknown                        : Unknown (*1)
(7) Disk status
    normal                             : Normal
    abnormal (abnormal disk type (*3)) : Abnormal
    unknown                            : Unknown (*1)

*1: Displayed when Mirroring Controller is stop state, the management network is
abnormal, or Mirroring Controller has failed or is unresponsive.

*2: The names of the DBMS processes in which the abnormality was detected are output. The
name has the following meaning: However, if multiple DBMS process issues are detected,
only the DBMS for which the first issue was detected is displayed.
     -postmaster: Process (postmaster) that accepts application connections
     -wal_sender or wal_receiver: Process (WAL sender or WAL receiver) that sends and
receives transaction logs
*3: The types of disks where the abnormality was detected are output separated by a
comma. The type has the following meaning:
     -data: Data storage disk
     -tran_log: Transaction log storage disk
     -tablespace: Tablespace storage disk
```

Display details of mc_ctl status (with the --arbiter option specified)

```
arbiter_id   host            status
(1)          (2)              (3)
```

```
(1) Arbitration server identifier
(2) Host name or IP address
(3) Mirroring Controller arbitration process connection status
    online  : Connected
    offline : Disconnected (*1)

*1: When Mirroring Controller is stopped, connections to the Mirroring Controller
arbitration process cannot be established, so it will be displayed as "offline".
```

Chapter 5 Connection Manager Commands

This chapter explains the Connection Manager commands.

5.1 cm_ctl

Name

cm_ctl - Start, stop or display the status of the conmgr process

Synopsis

```
cm_ctl start [-D diretory] [-W] [--complete] [-t seconds]

cm_ctl stop [-D diretory] [-W] [-m {smart | fast | immediate}] [-t seconds]

cm_ctl status [-D directory] [-t seconds] [-i {all | instance | application}]
```

Description

The start mode starts the conmgr process. The command returns at least after the heartbeat monitoring connection is completed with the primary server's instance.

When --complete is specified, wait until all instances configured in conmgr.conf have completed their heartbeat monitoring connections. You can set a timeout for these waits. The default of timeout is 60 seconds. Can be changed using the -t option. If it times out, it simply gives up waiting and the conmgr process remains up.

Note

If the primary is not among the instances configured in the cmgr.conf, use the -W option when starting Connection Manager with the cm_ctl command. Without the -W option, the cm_ctl command will not return until the primary connection is complete.

This includes, for example, starting up conmgr which is set up in advance to connect only to the standby center side of the disaster countermeasure operation.

The stop mode sends a signal to the conmgr process to shut down and wait until the process disappeares.

The default of wait time is 60 seconds. Can be changed using the -t option. If it times out, it simply gives up waiting. There are three shutdown methods, "smart", "fast", and "immediate", specify with the -m option. The "smart" waits until all applications using the conmgr process run out of SQL connections before shutting down. The "fast" forces all applications using the conmgr process to disconnect from the conmgr process before shutting it down. As a result, the SQL connection for the application receives an error. The "immediate" terminates the conmgr process immediately. If nothing is specified, it stops in smart mode.

The status mode, if the commgr process exists, queries the conmgr process for instance and application information known to the conmgr process, and display them to standard output along with the state of conmgr itself.

The -i option allows you to specify what information to query. The "instance" queries information about the instance; The "application" queries information about the application; The "all" queries information about both. conmgr's own information is always displayed. The default time to wait for a query to return is 60 seconds. Can be changed using the -t option.

Options

--complete

Wait until all instances configured in conmgr.conf have completed their heartbeat monitoring connections. If the When used with the -W option, the -W option takes precedence.

-D

-- directory=*directory*

Specify the directory where conmgr.conf is located. If omitted, it refers to the directory specified by the environment variable CMDATA. You cannot omit both.

-i {all | instance | application}

Specify the information to display the status.

-m

--mode={smart | fast | immediate}

Specify the mode of shutdown. The default is smart.

-t *seconds*

--timeout=*seconds*

Specify how long to wait for the operation to complete. The unit is seconds. The default is 60 seconds.

-W

--no-wait

In start mode, cm_ctl command returns immediately after forking conmgr process. In stop mode, the cm_ctl command returns without waiting for the process to disappeare.

Diagnostics

start mode or stop mode

0: Normal end

2: Timeout occurred

3: Unable to access the specified directory

Other than the above: None of the above

status mode

0: Normal end

2: Timeout occurred

3: Unable to access the specified directory

4: conmgr process does not exist

Other than the above: None of the above

Privileges

The conmgr process cannot be started by the administrator(e.g. superuser(root) on Linux).

Example

Display details of start mode

The block of information that can be specified with the -i option is used as a unit. There is one blank line between the blocks and no blank lines within the blocks. It includes one or more spaces between columns and between data.

```
$ cm_ctl status -i all
conmgr_status:
status pid
(1)     (2)
ready  3456

instance_information:
addr          port  database_attr
(3)           (4)   (5)
```

```
192.0.2.100   27500 standby
192.0.2.110   27500 primary
192.0.2.120   27500 standby
192.0.2.130   27500 unknown

application_infomation:
addr          port  pid   connected_time
(6)           (7)   (8)   (9)
127.0.0.1     12345 5678  2020-01-15 02:03:04
```

(1) Status of the conmgr process
    starting : The process is starting its startup sequence but is not ready to accept
               connections from clients.
    ready   : Ready to accept connections from clients.
    stopping : It has received an end instruction and is starting the stop sequence.
    inactive : The conmgr process does not exist.
(2) PID of the conmgr process
(3) IP address of the instance
(4) Port number the postmaster listens on
(5) Status of the instance (primary | standby | unknown)
(6) Connection source IP address for conmgr process
(7) Connection source (ephemeral) port number for conmgr process
(8) PID of the connection source
(9) Date and time conmgr process connection
    The ISO 8601 compliant date is followed by a blank, followed by the ISO 8601
    compliant second precision time.
    This representation is a PostgreSQL string representation of type timestamp.