FUJITSU

Fujitsu
Enterprise Postgres 16 SP1

Installation and Setup Guide for Client

Linux

J2UL-2950-02PEZ0(00)
September 2024

Preface

Purpose of this document

This document describes how to install, uninstall and set up the "Fujitsu Enterprise Postgres client feature".

Intended readers

This document is intended for those who install and operate Fujitsu Enterprise Postgres.

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

- PostgreSQL
- SQL
- Linux

Structure of this document

This document is structured as follows:

Chapter 1 Overview of Installation

Describes the features that can be installed, and provides an overview of installation methods

Chapter 2 Installation and Uninstallation of the Linux Client

Describes how to install the Fujitsu Enterprise Postgres client feature (Linux client)

Chapter 3 Setup

Describes the setup procedures to be performed after installation completes

Export restrictions

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

Issue date and version

```
Edition 2.0: September 2024
Edition 1.0: June 2024
```

Copyright

Contents

Chapter 1 Overview of Installation

This chapter provides an overview of Fujitsu Enterprise Postgres installation.

1.1 Features that can be Installed

Fujitsu Enterprise Postgres provides features to enable access to the database from a variety of platforms and languages, as the connection environment for the client and the database server.

The Fujitsu Enterprise Postgres client package must be installed on the client system to use these features.

The following list shows the features provided by client packages.

- JDBC
- ODBC
- C language (libpq)
- Embedded SQL (ECPG) in C language
- Connection Manager
- High speed data load
- Pgpool-II
- ldap2pg
- pgBackRest

1.2 Installation Types

The following installation types are available for Fujitsu Enterprise Postgres:

- New installation
- Reinstallation

1.2.1 New Installation

In initial installation, the Fujitsu Enterprise Postgres client feature is installed for the first time.

1.2.2 Reinstallation

Perform reinstallation to repair installed program files that have become unusable for any reason.

1.3 Uninstallation

Uninstallation removes the system files of the installed Fujitsu Enterprise Postgres client feature.

Chapter 2 Installation and Uninstallation of the Linux Client

This chapter explains how to install and uninstall the Linux client.

2.1 Operating Environment

This section describes the operating environment required to use the Linux client.

2.1.1 Required Operating System

The following operating systems is required to use the Linux client. Check and use minor version, which is certified and currently supported by Red Hat or SUSE for IBM Power LE (POWER9 and POWER10).

- RHEL8.4 or later minor version
- RHEL9.2 or later minor version
- SLES 15 SP3 or later minor version

Information

- The following packages are required for operations on RHEL8.

Package name	Remarks
bzip2-libs	Required when using pgBackRest.
glibc	-
libnsl2	-
libgcc	-
libmemcached	Required when using Pgpool-II.
libstdc++	-
libtool-ltdl	-
libzstd	-
ncurses-libs	-
nss-softokn-freebl	-
rsync	Required when using Pgpool-II.
unixODBC	Required when using ODBC drivers.
xz-libs	-
zlib	-

- The following packages are required for operations on RHEL9.

Package name	Remarks
bzip2-libs	Required when using pgBackRest.
glibc	-
libnsl2	-
libgcc	-
libmemcached	Required if Pgpool-II is used.
libstdc++	-

Package name	Remarks
libtool-ltdl	-
libzstd	-
ncurses-libs	-
nss-softokn-freebl	-
rsync	Required if Pgpool-II is used.
unixODBC	Required if you are using an ODBC driver.
xz-libs	-
zlib	-

- The following packages are required for operations on SLES 15.

Package name	Remarks
glibc	-
libaudit1	-
libbz2-1	Required when using pgBackRest.
libgcc_s1	-
libltdl7	-
libmemcached	Required when using Pgpool-II.
libncurses6	-
libstdc++	-
libz1	-
libzstd1	-
rsync	Required when using Pgpool-II.
unixODBC	Required when using ODBC drivers.

2.1.2 Related Software

The following table lists the software required to use the Linux client.

Table 2.1 Related software

No.	Software name	Version
1	C compiler (*1)	-
2	JDK or JRE (*2)	JDK 8 JRE 8 JDK 11 JRE 11 JDK 17 JRE 17

*1: Only operations using the C compiler provided with the operating system are guaranteed.

*2: OpenJDK is supported.

The following table lists servers that can be connected to the Linux client.

Table 2.2 Connectable servers

OS	Software name
Linux	Fujitsu Enterprise Postgres Advanced Edition 14 or later , up to 16 SP1

2.1.3 Excluded Software

There are no exclusive products.

2.1.4 Required Patches

There are no required patches.

2.1.5 Hardware Environment

The following hardware is required to use the Linux client.

Memory

At least 160 MB of memory is required.

Mandatory hardware

None.

2.1.6 Disk Space Required for Installation

The following table lists the disk space requirements of the corresponding directories for new installation of the Linux client. If necessary, increase the size of the file system.

Table 2.3 Disk space required for installation

Directory	Required disk space Unit: MB
/etc	1
Installation destination of the client	121
Installation destination of ldap2pg	30
Installation destination of pgBackRest	40

2.1.7 Supported System Environment

This section describes the supported system environment.

TCP/IP protocol

Fujitsu Enterprise Postgres supports version 4 and 6 (IPv4 and IPv6) of TCP/IP protocols.

Note

Do not use link-local addresses if TCP/IP protocol version 6 addresses are used.

2.1.8 Versions of Open-Source Software Used as the Base for Fujitsu Enterprise Postgres Drivers

For the version of open-source software that Fujitsu Enterprise Postgres each driver is based on, refer to "OSS Supported by Fujitsu Enterprise Postgres" in the General Description.

2.2 Installation

This section explains how to install the Linux client.

2.2.1 Pre-installation Tasks

Check the system environment for the following before the Linux client is installed.

Check the disk capacity

Check if sufficient free disk space is available for installing the Linux client.

Refer to "Table 2.3 Disk space required for installation" for information on disk space requirements.

If sufficient free disk space is unavailable, reconfigure disk partitions.

Executable Users

Installation and uninstallation is performed by superuser.

On the system, run the following command to become superuser.

```
$ su -
Password:******
```

2.2.2 Run Installation

The installation procedure is described below.

Note

The following characters can be used as input values:

Alphanumeric characters, hyphens, commas and forward slashes

1. Stop applications and programs

If the installation method is the following, all applications and programs that use the product must be stopped:

- Reinstallation

Before starting the installation, stop the following:

- Applications that use the product
- Connection Manager
- pgBadger
- Pgpool-II
- ldap2pg
- pgBackRest

2. Mount the DVD drive

Insert the client program DVD into the DVD drive, and then execute the following command:

Example

```
# mount -t iso9660 -r -o loop /dev/dvd /media/dvd
```

Here /dev/dvd is the device name for the DVD drive (which may vary depending on your environment), and /media/dvd is the mount point (which may need to be created before calling the command).

Note

If the DVD was mounted automatically using the automatic mount daemon (autofs), "noexec" is set as the mount option, so the installer may fail to start. In this case, use the mount command to remount the DVD correctly, and then run the installation. Note that the mount options of a mounted DVD can be checked by executing the mount command without any arguments.

3. Run the installation

Install the following packages (rpm) with the rpm command.

Feature Name	Operating System	Package (Path)
Client	RHEL8	CLIENT64/Linux/packages/r80ppc64le/FJSVfsep-CL-*.rpm
	RHEL9	CLIENT64/Linux/packages/r90ppc64le/FJSVfsep-CL-*.rpm
	SLES 15	CLIENT64/Linux/packages/SUSE15ppc64le/FJSVfsep-CL-*.rpm
Pgpool-II	RHEL8	PGPOOL2/Linux/packages/r80ppc64le/FJSVfsep-POOL2-*.rpm
	RHEL9	PGPOOL2/Linux/packages/r90ppc64le/FJSVfsep-POOL2-*.rpm
	SLES 15	PGPOOL2/Linux/packages/SUSE15ppc64le/FJSVfsep-POOL2-*.rpm
ldap2pg	RHEL8	LDAP2PG/Linux/packages/r80ppc64le/FJSVfsep-LD2PG-*.rpm
	RHEL9	LDAP2PG/Linux/packages/r90ppc64le/FJSVfsep-LD2PG-*.rpm
	SLES 15	LDAP2PG/Linux/packages/SUSE15ppc64le/FJSVfsep-LD2PG-*.rpm
pgBackRest	RHEL8	PGBACKREST/Linux/packages/r80ppc64le/FJSVfsep-PGBR-*.rpm
	RHEL9	PGBACKREST/Linux/packages/r90ppc64le/FJSVfsep-PGBR-*.rpm
	SLES 15	PGBACKREST/Linux/packages/SUSE15ppc64le/FJSVfsep-PGBR-*.rpm

*is the version, OS, etc.

Example

In the following example, /media/dvd is the name of the mount point where the DVD is mounted.

The "<*x*>" and " <*x*0*z*>" in the path indicate the x and z of the *x* SP*z* represented as the product version. For products without SP*z*, <*x*0*z*> becomes <*x*00>.

Below is an example of new installation on RHEL9.

```
# cd /media/dvd/CLIENT64/Linux/packages/r90ppc64le
# rpm -ivh FJSVfsep-CL-<x>-<x0z>-0.el9.ppc64le.rpm
```

Below is an example of new installation on SLES 15.

```
# cd /media/dvd/CLIENT64/Linux/packages/SUSE15ppc64le
# rpm -ivh FJSVfsep-CL-<x>-<x0z>-0.s15.ppc64le.rpm
```

Below is an example of reinstallation on RHEL9.

```
# cd /media/dvd/CLIENT64/Linux/packages/r90ppc64le
# rpm -ivh --replacepkgs FJSVfsep-CL-<x>-<x0z>-0.el9.ppc64le.rpm
```

Below is an example of reinstallation on SLES 15.

```
# cd /media/dvd/CLIENT64/Linux/packages/SUSE15ppc64le
# rpm -ivh --replacepkgs FJSVfsep-CL-<x>-<x0z>-0.s15.ppc64le.rpm
```

Note

If you perform reinstallation, and an installation prefix (in the --prefix option of the rpm command) was used for the new installation, you must use the same prefix.

2.3 Uninstallation

This section describes the procedure for uninstalling the Linux client.

Note

- Before uninstalling the product, close the product program and all applications that are using it.

2.3.1 Run Uninstallation

The uninstallation procedure is described below.

1. Stop applications and programs

Before starting the uninstallation, stop the following:

- Applications that use the product
- Connection Manager
- pgBadger
- Pgpool-II
- ldap2pg
- pgBackRest

2. Verifying Installation Features

Verify that the feature to be removed is installed by executing the following command.

Feature Name	Package Name
Client	FJSVfsep-CL-<*x*>

* Where *x* is a number indicating the version.

Example

```
# rpm -qi FJSVfsep-CL-16
```

3. Run the uninstallation

Run the following command.

Example

```
# rpm -e FJSVfsep-CL-16
```

The installation directory may remain after uninstallation. If it is not required, delete it.

Chapter 3 Setup

This chapter describes the setup procedures to be performed after installation completes.

3.1 Configuring Environment Variables

Configure the following environment variables when using client commands.

PATH environment variable

Add "*installationDirectory*/bin".

MANPATH environment variable

Add "*installationDirectory*/share/man".

PGLOCALEDIR environment variable

Add "*installationDirectory*/share/locale".

Examples of environment variable configurations are shown below.

Example

Note that "<*x*>" indicates the product version.

```
$ PATH=/opt/fsepv<x>client64/bin:$PATH ; export PATH
$ MANPATH=/opt/fsepv<x>client64/share/man:$MANPATH ; export MANPATH
$ PGLOCALEDIR=/opt/fsepv<x>client64/share/locale ; export PGLOCALEDIR
```

3.2 Setting Up and Removing OSS

This section explains how to set up OSS supported by Fujitsu Enterprise Postgres.

If you want to use OSS supported by Fujitsu Enterprise Postgres, follow the setup procedure.

If you decide not to use the OSS supported by Fujitsu Enterprise Postgres, follow the removing procedure.

Information

In this section, the applicable database that enables the features of each OSS is described as "postgres".

Refer to "OSS Supported by Fujitsu Enterprise Postgres" in the General Description for information on OSS other than those described below.

3.2.1 pgBackRest

3.2.1.1 Setting Up pgBackRest

1. Install pgBackRest.

   To use the pgbackrest command on the same host as the Fujitsu Enterprise Postgres server, install pgBackRest using the server program DVD. If you want to use the pgbackrest command on a different host than the Fujitsu Enterprise Postgres server, install pgBackRest using the client program DVD.

2. Set the environment variable PATH for pgBackRest.

   The pgBackRest material is stored under /opt/fsepv<x>pgbackrest. Set the environment variable PATH to the storage location/bin of the pgBackRest material to be used.

   ```
   $ export PATH=/opt/fsepv<x>pgbackrest/bin:$PATH
   ```

3. Perform pgBackRest setup.

   Refer to "User Guides" in the pgBackRest website (https://pgbackrest.org/) for details.

Note

- This feature is not available for instances created with WebAdmin. It is available only for operation using server commands.
- The pg_rman, pgx_dmpall, and pgx_rcvall commands cannot be used when using pgBackRest because of conflicting shell commands to set archive_command.

3.2.1.2 Removing pgBackRest

1. Sets parameters in the postgresql.conf file.

   Reverses the information specified during setup

2. Restart Fujitsu Enterprise Postgres.
3. If it was set to perform periodic backups, unset it.

3.2.1.3 Servers to which pgBackRest can Connect

The following table lists server that pgBackRest can connected to.

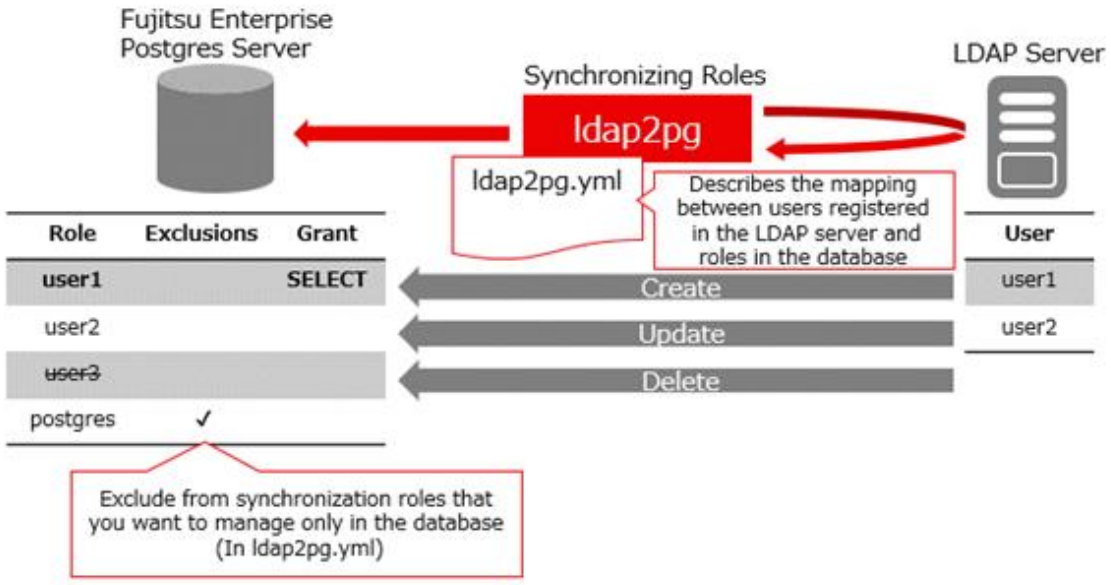
Table 3.1 Connectable server

OS	Product name
Linux	Fujitsu Enterprise Postgres Advanced Edition 16 or later

3.2.2 ldap2pg

PostgreSQL supports LDAP authentication and can be used on both Linux and Windows. You can use an LDAP server to authenticate users, but you must first create a role for the database server.

ldap2pg allows users registered with the LDAP server to be synchronized with Fujitsu Enterprise Postgres roles, so that the above database server roles can be created automatically. This allows you to centrally manage roles on the LDAP server. Note that ldap2pg only supports Linux.



Users registered with the LDAP server and Fujitsu Enterprise Postgres roles are synchronized when the ldap2pg command is executed, based on the ldap2pg.yml that defines these mappings. If a role defined in ldap2pg.yml does not exist in Fujitsu Enterprise Postgres, it is created, and any roles not defined in ldap2pg.yml are removed. Roles that would be difficult to update or delete, such as database administrator roles that do not work with LDAP servers, can be excluded from synchronization by setting them to ldap2pg.yml.

The key points of operation are explained below.

Timing of Synchronization

Synchronize when the LDAP server user changes so that the database server is always up to date. Therefore, you must synchronize periodically to automatically propagate the LDAP server information, or manually propagate it as the LDAP server changes.

If you synchronize periodically, ensure that the synchronization interval is an acceptable time lag before LDAP server changes are propagated to the database server. This is because, even when fully synchronized, ldap2pg accesses the LDAP server and database to check for changes. For example, run the ldap2pg command periodically every 5 minutes or so.

If you use cron, for example, to run automatically on a regular basis, you should log the standard output and standard error output of ldap2pg using settings or redirects such as cron. You can check the log to see if ldap2pg was interrupted or if an unexpected role was removed.

If you want to synchronize immediately or if you want to control the synchronization timing yourself, synchronize manually.

Enhanced Security in Combination with Confidentiality Management

ldap2pg can also manage database privileges, but it cannot manage granular units such as tables and rowsets. Combined with the confidentiality management, which allows such configuration and allows auditing of privilege settings, it provides robust security measures.

For the settings for using ldap2pg in combination with the confidentiality management, refer to "3.2.2.4 Configuration with Confidentiality Management".

3.2.2.1 Setting Up ldap2pg

1. Install ldap2pg
   Install ldap2pg using the client program DVD.
2. Set the environment variable PATH for ldap2pg.

```
$ export PATH=/opt/fsepv<x>ldap2pg/bin:$PATH
```

3. Define a database role on the database server that has superuser privileges as the executor of ldap2pg. For more information about defining roles, refer to "CREATE ROLE" in "Reference" in the PostgreSQL Documentation for information on the CREATE ROLE.
4. Perform ldap2pg setup.
   Refer to "Configuration" or "Cookbook" in the ldap2pg document (https://ldap2pg.readthedocs.io/en/latest/) for details.
5. Set roles that are defined and used only by the database, such as database administrators not managed by an LDAP server, or roles that exclude synchronization, as defined by Fujitsu Enterprise Postgres.
   Add the settings to roles_blacklist_query in the ldap2pg.yml file.

   Fujitsu Enterprise Postgres-specific roles to add:

   - pgx_update_profile_status, and roles that inherit from pgx_update_profile_status (Role for streaming replication of the Policy-based Login Security)
   - pgx_cgroup_role_* (Confidentiality role for the confidentiality management)

When the Database Server is redundant

In a database redundancy environment, specify "primary" for the target_session_attrs parameter. You can also specify "read-write".

3.2.2.2 Removing ldap2pg

1. If you have set ldap2pg to run periodically, unset it.
2. Uninstall ldap2pg. Refer to "2.3 Uninstallation" for more information.

3. If you have defined a role on the database server specifically for running ldap2pg, remove that role.

3.2.2.3 Using ldap2pg to Synchronize Database Roles

Describes how to use ldap2pg to synchronize users of an LDAP server with a database server as database roles.

1. Edit the ldap2pg.yml file, for example if you want to grant access to a role that synchronizes with an LDAP user. For information on ldap2pg.yml, refer to the following document:
   https://ldap2pg.readthedocs.io/en/latest/config/
2. Use environment variables to specify information about the connection destination to the LDAP server or database.
   https://ldap2pg.readthedocs.io/en/latest/cli/#environment-variables
   The user who connects to the database server must be the user created during the setup procedure. Connections to LDAP servers support LDAP-initiated environment variables and ldaprc files, while database access supports PG-initiated environment variables available in libpq. These environment variables are used to configure the connection.
3. Run ldap2pg with the check option to verify that the role being modified matches the role being modified.
4. Run ldap2pg with the --real option to synchronize roles with the database server.
5. Configure LDAP server users and database roles to synchronize periodically after the initial synchronization.
   Prepare the script that sets the environment variables and the script that synchronizes the roles that you performed in steps 2 and 4, and register the script in the cron job so that the script that synchronizes the roles references the environment variables and synchronizes the roles.

[Configuration Examples for cron]

```
SHELL=/bin/bash
*/5 * * * * source /home/postgres/env.sh && . /home/postgres/sample.sh >> /home/postgres/sample.log
2>&1
```

3.2.2.4 Configuration with Confidentiality Management

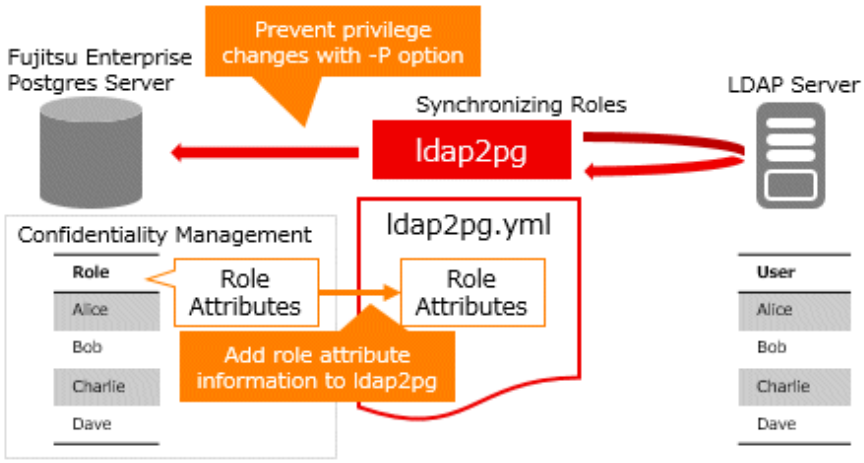
It combines ldap2pg with confidentiality management to provide detailed access control. There is overlap between the role management capabilities of ldap2pg and the confidentiality management. When used in combination, use ldap2pg and confidentiality management to separate role management:

Feature	Role Management Segregation
ldap2pg	Add, remove, and managing role membership
Confidentiality Management	Set role attributes, grant and revoke privileges, and audit them

To separate role management between ldap2pg and confidentiality management, do the following:

- Add attribute information for confidentiality management roles to the ldap2pg configuration file (ldap2pg.yml) so that the attributes of roles set for confidentiality management are not updated by running ldap2pg

- Run ldap2pg with the -P option to prevent deletion of confidentiality groups and role relationships in the confidentiality management when ldap2pg is run.



The configuration flow for ldap2pg combined with the confidentiality management is as follows.

Introduction

Configure the necessary settings to run ldap2pg as described in the following procedure.

1. Design user-role mappings on the LDAP server to create a list of roles that should be managed by the confidentiality management.
2. To create an yml file:
   a. Specify the settings for retrieving and synchronizing the listed objects from the LDAP server.
   b. Write a confidentiality management role starting with pgx_cgroup_role_ in roles_blacklist_query.
   c. Ensure that the grant and revoke privileges settings are not listed in the yml file.
3. Stop synchronization if it is already running using ldap2pg.
4. Create a role as described in "3.2.2.3 Using ldap2pg to Synchronize Database Roles".
5. Refer to "Confidentiality Management" in the "Security Operations Guide" and perform all necessary tasks.
   During this process, all the roles in the list of roles are registered in the confidentiality groups of the confidentiality management.
6. Modify the yml file so that ldap2pg does not update the attributes of the roles you have confidentiality management. Refer to "Settings When You Change the Attributes or Privileges of a role in a Confidentiality Groups" for a sample script that prints an yml file.
7. If you have already done regular synchronization using ldap2pg, try again.

Operation

Use the following procedure to manipulate roles according to your situation.

Adding an ldap2pg Role to a Confidentiality Groups

1. Creates a confidentiality management confidentiality groups.
2. Run ldap2pg with the -P option to create the LDAP server user as a database role.
3. Add the role you added above to the confidentiality groups.
4. Reflect the confidentiality management configuration in ldap2pg.yml, referring to the "Example of Applying Role Attributes".

Example of Applying Role Attributes

1. Use the following example to execute SQL and retrieve the settings for each role:
   For all roles, this example retrieves the LOGIN attribute, the role attributes of the confidentiality management, and the

membership of the confidentiality management role. If you want to change the settings to suit your environment, rewrite the SQL, such as modifying the 'LOGIN' part of the SQL Execution Example, or modify the Example of Run Results directly.

[SQL Execution Example]

```
SELECT '- name: ' ||pgxgr.name || chr(10) || '  options: ' || pgxgr.opt || chr(10) ||  '
parent: ' || chr(10) || '    - '  || string_agg(pgxgr.cgrorolename, chr(10) || '    - ')
FROM  (SELECT pgxg.cgrorolename,
        concat_ws(' ',
        'LOGIN',
        CASE pgxg.cgrosuperuser WHEN true THEN 'SUPERUSER' END,
        CASE pgxg.cgrocreatedb WHEN true THEN 'CREATEDB' END,
        CASE pgxg.cgrocreaterole WHEN true THEN 'CREATEROLE' END,
        CASE pgxg.cgroreplication WHEN true THEN 'REPLICATION' END,
        CASE pgxg.cgrobypassrls WHEN true THEN 'BYPASSRLS' END) AS opt,
        pgxroles.name
      FROM pgx_confidential_group pgxg,
      (SELECT pgxr.crolmatid as matid, pgxr.crolgroid as groid,  pgxr.crolname AS name
FROM pgx_confidential_role pgxr ) as pgxroles
      WHERE pgxg.cgromatid = pgxroles.matid and pgxg.cgroid = pgxroles.groid) pgxgr
GROUP BY pgxgr.name, pgxgr.opt;
```

[Example of Run Results]

```
- name: alice
    options: LOGIN CREATEDB
   parent:
    - pgx_cgroup_role_000000000000000001
- name: bob
   options: LOGIN CREATEDB
   parent:
    -pgx_cgroup_role_000000000000000001
- name: charlie
   options: LOGIN CREATEDB CREATEROLE
   parent:
    - pgx_cgroup_role_000000000000000002
- name: dave
   options: LOGIN CREATEDB CREATEROLE
   parent:
    - pgx_cgroup_role_000000000000000002
```

2. Put the setting of roles at the top of the rules in ldap2pg.yml based on the information in the above settings. If it is not at the top, the configuration information that synchronizes with the LDAP server takes effect, and the confidentiality management configuration does not take effect.

Settings When You Change the Attributes or Privileges of a role in a Confidentiality Groups

1. Confidentiality management modifies role attributes and privileges information.

2. Create a script to retrieve the confidentiality management configuration information and register it in a cron job so that the changed information is automatically reflected in the yml file.
   The following is an example shell script:.
   Please change the settings to suit your environment.
   The shell script shown here consists of two configuration files, ldap2pg_pre.yml and ldap2pg_after.yml, and the confidentiality management configuration information (In the sample, it is output to confidential_roles.yml) that is reflected in yml. Combine these three files to create the ldap2pg.yml file.
   ldap2pg_pre.yml is the information to be placed before the confidentiality management configuration information in ldap2pg.yml, and contains the postgres section and up to "roles:" in the rules section. ldap2pg_after.yml is information to be placed after tthe

confidentiality management configuration information in ldap2pg.yml, and contains information about roles not managed by the confidentiality management.

[Example of Shell Script]

ldap2pg_pre.yml : Provides information about the postgres section

```
version: 6

#
#       1.   P O S T G R E S   I N S P E C T I O N
#
#  See https://ldap2pg.readthedocs.io/en/latest/postgres/
#
postgres:
# Exclude roles starting with postgres, pg that PostgreSQL uses internally
  roles_blacklist_query: [postgres, pg_*, pgx_update_profile_status, pgx_cgroup_role* ]
  databases_query: [postgres]
(Omitted)
rules:
- description: "Setup static roles and grants."
  roles:
```

ldap2pg_after.yml : Provides information about roles that are not part of the confidentiality groups

```
  - names:
    - readers
    options: NOLOGIN
  - name: writers
    # Grant reading to writers
    parent: [readers]
    options: NOLOGIN
(Omitted)
```

sample.sh : A script that outputs information about confidentiality groups to confidential_roles.yml and combines them into a single yml file

```
#!/bin/bash

psql -h localhost -p 27500 -d postgres -U postgres -A -t <<EOF > /home/postgres/
confidential_roles.yml
SELECT '  - name: ' ||pgxgr.name || chr(10) || '    options: ' || pgxgr.opt || chr(10) ||  '
parent: ' || chr(10) || '    - '  || string_agg(pgxgr.cgrorolename, chr(10) || '    - ')
FROM  (SELECT pgxg.cgrorolename,
        concat_ws(' ',
        'LOGIN',
        CASE pgxg.cgrosuperuser WHEN true THEN 'SUPERUSER' END,
        CASE pgxg.cgrocreatedb WHEN true THEN 'CREATEDB' END,
        CASE pgxg.cgrocreaterole WHEN true THEN 'CREATEROLE' END,
        CASE pgxg.cgroreplication WHEN true THEN 'REPLICATION' END,
        CASE pgxg.cgrobypassrls WHEN true THEN 'BYPASSRLS' END) AS opt,
        pgxroles.name
      FROM pgx_confidential_group pgxg,
      (SELECT pgxr.crolmatid as matid, pgxr.crolgroid as groid,  pgxr.crolname AS name FROM
pgx_confidential_role pgxr ) as pgxroles
      WHERE pgxg.cgromatid = pgxroles.matid and pgxg.cgroid = pgxroles.groid) pgxgr
GROUP BY pgxgr.name, pgxgr.opt;
EOF
cat /home/postgres/ldap2pg_pre.yml /home/postgres/confidential_roles.yml /home/postgres/
ldap2pg_after.yml > /home/postgres/ldap2pg.yml
```

```
#Run  ldap2pg -P -c ldap2pg.yml to update retrieved role information
```

Information

If you want to manually apply the attribute or privilege information of a role that has been changed in confidentiality management to ldap2pg.yml, obtain the change information and apply it to ldap2pg.yml, referring to "Example of Applying Role Attributes".

Adding Roles Created with ldap2pg to a Confidentiality Groups

1. Create a role to add to the confidentiality groups in ldap2pg.
2. Add the database role you created in step 1 to the existing confidentiality groups.
3. Reflect the newly added role's confidentiality management settings in ldap2pg.yml, as shown in "Example of Applying Role Attributes".

Information

If cron automatically reflects changes to the confidentiality groups in ldap2pg.yml, stop cron and add the newly added database role to the confidentiality groups.

Removing Roles Added in ldap2pg from a Confidentiality Groups

1. Remove the role you want to remove from the confidentiality groups.
2. Reflect changes to confidentiality management in ldap2pg.yml, referring to "Example of Applying Role Attributes".
3. Execute ldap2pg with the -P option to reflect.

Point

If you deleted the confidentiality matrix and the confidentiality groups, perform steps 2 and 3 above.

See

- If you accidentally delete a role managed by confidentiality management using the ldap2pg, refer to "How to Check Confidentiality Objects and Roles" in the Security Operation Guide to recover the role managed by confidentiality management.
- If you accidentally delete the confidentiality role in ldap2pg, refer to "Creating a Confidentiality Management Role" in the Security Operations Guide to recover.

3.2.2.5 Servers to which ldap2pg can Connect

The following table lists server that ldap2pg can connected to.

Table 3.2 Connectable server

OS	Product name
Linux	Fujitsu Enterprise Postgres Advanced Edition 16 or later

Index