Database Setup Guide for the IM and Presence Service, Release 12.5(1)

First Published: 2018-11-19

Last Modified: 2020-08-28

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

CHAPTER 1

External Database Requirements

This guide provides information about how to configure an external database for Cisco Unified Communications Manager IM and Presence Service features. The following features require an external database:

- Persistent Group Chat
- High Availability for Persistent Chat
- Message Archiver (IM Compliance)
- Managed File Transfer

How to use this Guide

Refer to the following chapters for instructions on how to configure your external database.

Procedure

	Command or Action	Purpose
Step 1	External Database Requirements, on page 1	Review support information and other requirements for your external database.
Step 2	Install the external database: • Install PostgreSQL, on page 7 • Install Oracle, on page 11 • Install Microsoft SQL Server, on page 15	Refer to one of the chapters on the left for installation information.
Step 3	Configure IM and Presence Service for External Database, on page 23	Configure the IM and Presence Service for the external database connection.

What to do next

After setting up the external database, refer to the additional material in this guide for information on administering your external database.

External Database Setup Requirements

General Requirements

Cisco suggests having a certified PostgreSQL, Oracle, or Microsoft SQL Server administrator to maintain and retrieve information from the external database.

Hardware and Networking Requirements

- The external database should be installed on a dedicated server. The database can be deployed on virtualized or non-virtualized platforms.
- See the vendor database documentation for details on supported operating systems and platform requirements.
- IPv4 and IPv6 are supported by IM and Presence Service.

Software Requirements

The following table contains general external database support information for the IM and Presence Service. For detailed information specific to IM and Presence features, refer to the subsequent "Feature Requirements" section.

Table 1: Database Support for the IM and Presence Service

Database	Supported Versions
PostgreSQL	**Note** • PostgreSQL 12.x and higher are compatible only with IM and Presence Service Release, 12.5(1) SU6 and higher. All currently active PostgreSQL versions, from 12.x through 16.x are supported. At this time, validation was performed against versions 12.x, 13.x and 16.x.
Oracle	Testing is performed using Oracle 9g, 10g, 11g, 12c, and 19c versions. Since the IM and Presence features are using common Oracle features such as basic SQL statements, Stored Procedures, and basic indexing; we expect that future versions remain compatible and will be supported unless otherwise specified in this document. Cisco plans to include compatibility testing of newer major Oracle DB releases during future major IM and Presence releases.
Microsoft SQL Server	Testing is performed using MS SQL 2012, 2014, 2016, 2017, and 2019 versions. The IM and Presence features use common MS SQL features. The future releases and patches remain compatible unless otherwise specified in this document. Cisco plans to include compatibility testing of newer major DB releases during future major IM and Presence releases.

Feature Requirements

IM and Presence features which depend on the integration with an external database support Oracle, PostgreSQL, and Microsoft SQL Server as external databases. External database requirements differ depending on which features you want to deploy on the IM and Presence Service. Refer to the following table for support information for specific IM and Presence features.

Table 2: External Database Requirements for Specific IM and Presence Features

Feature	Requirements
Persistent Group Chat feature	A minimum of one unique logical external database instance (tablespace) is required for the entire IM and Presence Service intercluster. A unique logical external database instance for each IM and Presence Service node or redundancy group in an IM and Presence Service cluster will provide optimum performance and scalability, but is not mandatory.
High Availability for Persistent Chat feature	Make sure that both presence redundancy group nodes are assigned to the same unique logical external database instance. Note that Cisco does not provide detailed back-end database support. Customers are responsible for resolving back-end database issues on their own.
Message Archiver (compliance) feature	We highly recommend that you configure at least one external database for each IM and Presence Service cluster; however you may require more than one external database for a cluster depending on your database server capacity.
Managed File Transfer feature	You require one unique logical external database instance for each IM and Presence Service node in an IM and Presence Service cluster. **Note** Database table space can be shared across multiple nodes or clusters provided capacity and performance isn't overloaded.

Note If you deploy any combination of the persistent group chat, message archiver (compliance), and managed file transfer features on an IM and Presence Service node, the same unique logical external database instance (tablespace) can be shared across the features as each feature uses separate data tables. This is dependent on the capacity of the database instance.

Additional Documentation

This procedure only describes how to configure the external database on the IM and Presence Service. It does not describe how to fully configure the features that require an external database. See the documentation specific to the feature you are deploying for the complete configuration:

- For information on configuring the message archiver (compliance) feature on the IM and Presence Service, see *Instant Messaging Compliance for IM and Presence Service.*
- For information on configuring the persistent group chat feature on the IM and Presence Service, see *Configuration and Administration of the IM and Presence Service.*

- For information on configuring the managed file transfer feature on the IM and Presence Service, see *Configuration and Administration of the IM and Presence Service* .

External Database Setup Prerequisites

Before you install and configure the external database on the IM and Presence Service, perform the following tasks:

- Install the IM and Presence Service nodes as described in *Installation Guide for Cisco Unified Communications Manager and the IM and Presence Service*.
- Configure the IM and Presence Service nodes as described in *Configuration and Administration of IM and Presence Service*.

⚠

Caution If the IM and Presence Service connects to an external database server using IPv6, ensure that the enterprise parameter is configured for IPv6 and that Eth0 is set for IPv6 on each node in the deployment; otherwise, the connection to the external database server fails. The message archiver and Cisco XCP Text Conference Manager will be unable to connect to the external database and will fail. For information about configuring IPv6 on the IM and Presence Service, see *Configuration and Administration of IM and Presence Service*.

Performance Considerations

When you configure an external database with the IM and Presence Service, you must consider the following recommendations:

- Reduce the round-trip delay (RTT) between the IM and Presence Service cluster and the external database to avoid performance issues. This is normally accomplished by locating the external database server as close as possible to the IM and Presence Service cluster.
- Do not allow the external database entries to be full which causes performance issues on the IM and Presence Service cluster. Regular maintenance of the external database plays an important role in preventing IM and Presence Service performance degradation.

Note The external database maintenance further tunes the query execution mechanisms of the database engine itself when the number of records in the database reaches certain threshold.

For example, on the MSSQL database by default when you enable the query execution optimization mechanism that is called Parameter Sniffing, it can negatively affect the performance of persistent chat service. If this optimization mechanism does not adjust with plan guides for concrete IM and Presence Service queries, delay in Instant Messages delivery to persistent chats will be introduced.

Related Topics

PostgreSQL documentation
Oracle documentation
Microsoft Server documentation

About Security Recommendations

External Database Connection Security

The IM and Presence Service provides a secure TLS/SSL connection to the external database but only when Oracle or Microsoft SQL Server is chosen as the database type. We recommend that you consider this security limitation when you plan your IM and Presence Service deployment, and consider the security recommendations we provide in this topic.

Maximum Limit Connection Setup

For additional security, you can limit the maximum number of permitted connections to the external database. Use the guideline we provide here to calculate the number of database connections that are appropriate for your deployment. This section is optional configuration. The guideline infers that:

- You are running the managed file transfer, message archiver (compliance), and persistent group chat features on the IM and Presence Service.
- You configure the default number of connections to the database for the persistent group chat feature on the **Cisco Unified CM IM and Presence Administration** interface.

Guideline

PostgreSQL — max_connections = (N ×15) + Additional Connections

Oracle — QUEUESIZE = (N ×15) + Additional Connections

Microsoft SQL Server — the maximum number of concurrent connections = (N x15) + Additional Connections

- N is the number of nodes in your IM and Presence Service cluster.
- 15 is the default number of connections to the database on the IM and Presence Service, that is, five connections each for the managed file transfer, message archiver, and persistent group chat features.
- Additional Connections represents any independent administration or database administrator (DBA) connections to the database server.

PostgreSQL

To limit the number of PostgreSQL database connections, configure the `max_connections` value in the `postgresql.conf file` located in the *`install_dir`*`/data` directory. We recommend that you set the value of the `max_connections` parameter equal to, or slightly larger than, the above guideline.

For example, if you have an IM and Presence Service cluster containing six nodes, and you require an additional three DBA connections, using the guideline above, you set the max_connections value to 93.

Oracle

To limit the number of Oracle database connections, configure the QUEUESIZE parameter in the listener.ora file located in the *`install_dir`*`/data` directory. We recommend that you set the value of the QUEUESIZE parameter equal to the above guideline.

For example, if you have an IM and Presence Service cluster containing 4 nodes, and you require one additional DBA connection, using the guideline above, you set the QUEUESIZE value to 61.

Microsoft SQL Server

To limit the number of MS SQL Server database simultaneous connections carry out the steps below. We recommend that you set the size of the queue equal to the above guideline.

1. From the **SQL Server Configuration Manager**, right-click the node you want to configure and click **Properties**.
2. Click **Connections**.
3. In the **Connections** pane, enter a value from 0 to 32767 in the **Max number of concurrent connections** dialog box.
4. Restart the Microsoft SQL Server.

Default Listener Port Setup

Note This section is an optional configuration.

For additional security, you may choose to change the default listening port on the external database:

- For PostgreSQL, see Set Up PostgreSQL Listening Port, on page 9 for details on how to edit the default listener port.
- For Oracle, you can edit the default listener port by editing the `listener.ora` config file
- For Microsoft SQL Server, you can assign a TCP/IP port number as the default listener port in the SQL Server Configuration Manager. For details, see Default Listener Port Setup for Microsoft SQL Server, on page 19.

CHAPTER 2

Install PostgreSQL

This chapter provides information about installing and setting up PostgreSQL.

Install PostgreSQL Database

Before you begin

- Cisco recommends that a PostgreSQL DBA install and maintain the PostgreSQL server.
- Read the security recommendations for the PostgreSQL database in section About Security Recommendations, on page 5.
- For information on supported versions, see External Database Setup Requirements, on page 2.

Procedure

Step 1 Enter these commands to sign in to the database server as a Postgres user:

```
>su - postgres
```

```
>psql
```

Step 2 Create a new database user. The example below creates a new database user called *tcuser*:

```
#CREATE ROLE tcuser LOGIN CREATEDB;
```

Note If you deploy PostgresSQL version 8.4.x, you must configure the database user as a superuser at this point in the procedure, for example:

```
#ALTER ROLE tcuser WITH SUPERUSER;
```

Step 3 Create the database. If your database contains ASCII characters only, create the database with SQL_ASCII encoding. If your database contains non-ASCII characters, create the database with UTF8 encoding.

The example below creates an SQL_ASCII database called *tcmadb*.

```
#CREATE DATABASE tcmadb WITH OWNER tcuser ENCODING 'SQL_ASCII';
```

Step 4 Configure user access to the database. Edit the `install_dir/data/pg_hba.conf` file to allow the *postgres* user and the new *tcuser* user to access the database. For example:

# TYPE	DATABASE	USER	CIDR-ADDRESS	METHOD
host	tcmadb	tcuser	10.89.99.0/24	password
host	dbinst	mauser	10.89.99.0/24	password
local[1]	all	all		Trust or MD5

[1] **For Unix domain socket connections only.**

Step 5 Enter these commands to define passwords for the *postgres* and *tcuser* users:

```
#ALTER ROLE postgres WITH PASSWORD 'mypassword';
```

```
#ALTER ROLE tcuser WITH PASSWORD 'mypassword';
```

Note You are required to enter a password for the database user when you configure an external database entry on the IM and Presence Service.

Step 6 If you are running the PostgreSQL version 8.3.7 or a later 8.3.x release, change the permission of the *tcuser* to superuser to allow this user access to the database. Enter this command:

```
#ALTER ROLE tcuser WITH SUPERUSER;
```

Step 7 Configure the connections to the database from remote hosts. Edit the listen_addresses parameter in the `install_dir/data/postgresql.conf file`. For example:

```
listen_addresses = '*'
```

Step 8 If you are running PostgreSQL version 9.1.1, or higher, you must set the following values in the `postgresql.conf` file:

```
escape_string_warning = off
```

```
standard_conforming_strings = off
```

Step 9 Stop and restart the PostgreSQL service, for example:

```
/etc/rc.d/init.d/postgresql-8.3 stop
```

```
/etc/rc.d/init.d/postgresql-8.3 start
```

Note The commands to stop and start the PostgreSQL service may vary between PostgreSQL releases.

Step 10 Enter these commands to sign in to the new database as the *postgres* user and enable PL/pgSQL:

```
>psql tcmadb -U postgres
```

Note The following example, up to the semicolon, should be entered as one line.

```
#CREATE FUNCTION plpgsql_call_handler () RETURNS LANGUAGE_HANDLER AS '$libdir/plpgsql'
LANGUAGE C;
```

```
#CREATE TRUSTED PROCEDURAL LANGUAGE plpgsql HANDLER plpgsql_call_handler;
```

Troubleshooting Tips

Do not turn on the following configuration items in the *`install_dir`*`/data/postgresql.conf` file (by default these items are commented out):

```
client_min_messages = log
```

```
log_duration = on
```

Related Topics

About Security Recommendations, on page 5

Set Up PostgreSQL Listening Port

Note This section is optional configuration.

By default, the Postgresql database listens on port 5432. If you want to change this port, you must edit the PGPORT environment variable in /etc/rc.d/init.d/postgresql with the new port number.

Note The PGPORT environment variable overrides the 'Port' parameter value in the `/var/lib/pgsql/data/postgresql.conf` file, so you must edit the PGPORT environment variable if you want the Postgresql database to listen on a new port number.

Procedure

Step 1 Edit the PGPORT environment variable in /etc/rc.d/init.d/postgresql with the new port, for example:

```
IE: PGPORT=5555
```

Step 2 Enter these commands to stop and start the PostgreSQL service:

```
# /etc/rc.d/init.d/postgresql start
```

```
# /etc/rc.d/init.d/postgresql stop
```

Step 3 Confirm that the Postgresql database is listening on the new port using this command:

```
'lsof -i -n -P | grep postg'
```

```
postmaste 5754 postgres 4u IPv4 1692351 TCP *:5555 (LISTEN)
```

Tip For IPv6 servers, enter `postmaste 5754 postgres 4u IPv6 1692351 TCP *:5555 (LISTEN)`

Step 4 To connect to the database after you have changed the port, you must specify the new port number in the command using the -p argument. If you do not include the -p argument in the command, the Postgresql database attempts to use the default port of 5432, and the connection to the database fails.

For example:

```
psql tcmadb -p 5555 -U tcuser
```

User Access Restriction Recommendations

We strongly recommend that you restrict user access to the external database to only the particular user and database instance that the IM and Presence Serivce uses. You can restrict user access to the PostgreSQL database in the pg_hba.conf file located in the <install_dir>/data directory.

Caution Do not configure 'all' for the user and database entries because potentially this could allow any user access to any database.

When you configure user access to the external database, we also recommend that you configure password protection for the database access using the 'password' method.

Note You are required to enter a password for the database user when you configure a database entry on IM and Presence Service.

The following are examples of a secure user access configuration, and a less secure user access configuration, in the pg_hba.conf file.

Example of a secure configuration:

# TYPE	DATABASE	USER	CIDR-ADDRESS	METHOD
host	dbinst1	tcuser1	10.89.99.0/24	password
host	dbinst2	mauser1	10.89.99.0/24	password

Example of a less secure configuration:

# TYPE	DATABASE	USER	CIDR-ADDRESS	METHOD
host	dbinst1	tcuser1	10.89.99.0/24	trust
host	dbinst2	all	10.89.99.0/24	password

Notes on the example of a less secure configuration:

- The first entry contains no password protection for the database.
- The second entry allows any user to access the database “dbinst2”.

Related Topics

Install PostgreSQL Database, on page 7

PostgreSQL documentation

CHAPTER 3

Install Oracle

This chapter provides information about installing and setting up an Oracle database.

Install Oracle Database

Before you begin

- Cisco recommends that an Oracle DBA install the Oracle server.
- You need to update the patch for the known Oracle defect: ORA-22275. If this is not done persistent chat rooms will not work properly.
- Read the security recommendations for the Oracle database in your Oracle documentation.
- For information on supported versions, see External Database Setup Requirements, on page 2.
- For Oracle version 11 and earlier, you must configure your Oracle database to use UTF8 character encoding.
- As of Oracle version 12, you must configure the Oracle database to use AL32UTF8 character encoding, as UTF8 may lead to unexpected behavior. For example, if you use UTF8 with Oracle 12, chat rooms may be deleted when you restart the Cisco XCP Text Conference Manager service.
- To install the Oracle database, refer to your Oracle documentation.

To create tablespace and a database user, connect to the Oracle database as sysdba:

```
sqlplus / as sysdba
```

Procedure

Step 1 Create tablespace.

Note The **DATAFILE** keyword of the **CREATE TABLESPACE** command tells Oracle where to put the tablespace's datafile.

a) Enter the following command:

```
CREATE TABLESPACE tablespace_name DATAFILE
'absolute_path_to_oracle_installation\oradata\database_name\datafile.dbf' SIZE 100M
AUTOEXTEND ON NEXT 1M MAXSIZE UNLIMITED LOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE
MANAGEMENT AUTO;
```

- Replace *tablespace_name* with the tablespace name.
- Replace *absolute_path_to_oracle_installation* with the absolute path to where Oracle is installed. The entire path, including *datafile*.dbf, is enclosed in single quotation marks.
- Replace *database_name* with the name of your database folder.
- The *datafile*.dbf must be created in a folder under `\oradata\`, in this case the *database_name* folder.
- Replace *datafile*.dbf with the datafile name you want to create.

Step 2 Create a database user.

```
CREATE USER user_name IDENTIFIED BY "new_user's_password" DEFAULT TABLESPACE tablespace_name
TEMPORARY TABLESPACE "TEMP" QUOTA UNLIMITED ON tablespace_name ACCOUNT UNLOCK;
```

- Replace *user_name* with the new user's user name.

  Note The command `CREATE USER user_name` without double quotes will default to upper case and with quotes it will maintain the case

- Replace "*new_user's_password*" with the new user's password.

  Important Enclosing the *new_user's_password* within double quotation marks makes the variable case-sensitive. By default SQL identifiers are not case-sensitive.

- Replace *tablespace_name* with the tablespace name.

Step 3 Grant permissions to the database user.

The following example grants the required permissions and privileges to a database user, which are needed to create or upgrade the schema:

Note Prior to an upgrade, you must ensure that these permissions and privileges are granted, so that all IM and Presence Service services continue to operate as normal following the upgrade.

- `GRANT CREATE SESSION TO  user_name;`
- `GRANT CREATE TABLE TO  user_name;`
- `GRANT CREATE PROCEDURE TO  user_name;`
- `GRANT CREATE TRIGGER TO user_name;`

After you have created or upgraded the schema, the following privileges can be revoked if greater access control is required:

Note Ensure that revoked privileges are granted again before upgrading.

- `REVOKE CREATE TABLE FROM user_name;`
- `REVOKE CREATE PROCEDURE FROM user_name;`
- `REVOKE CREATE TRIGGER FROM user_name;`

Note IM and Presence Service only requires the `CREATE SESSION` privilege for regular operation.

Related Topics

Oracle Documentation

Create New Database Instance

Procedure

Step 1 Enter the command `dbca`
The **Database Configuration Assistant** wizard opens.

Step 2 Click **Next**.
The **Operations** window appears.

Step 3 Click the **Create a Database** radio button and then click **Next**.
The **Database Templates** window appears.

Step 4 Click the **General Purpose or Transaction Processing** radio button and then click **Next**.
The **Database Identification** window appears.

Step 5 Enter a unique Global Database Name on this screen and also a unique Oracle System Identifier (SID) for the database and click **Next**.

Note Take note of the SID because it is needed in Step 15.

The **Management Options** window appears.

Step 6 Under the Enterprise Manager tab the required settings are enabled by default but you can configure optional backups and alert notifications. Click **Next**.
The **Database Credentials** window appears.

Step 7 The window has two options to set up password authentication for database users, choose one and click **Next**.
The **Database File Locations** window appears.

Step 8 The Storage Type drop-down list should be the same as your Oracle Installation. Click the **Use Oracle-Managed Files** radio button and click **Next**.

Note This creates the new database instance in the same folder as your other database instances.

The **Recovery Configuration** window appears.

Step 9 Leave the default values and click **Next**.
The **Database Content** window appears.

Step 10 [Optional] Check the check box if you want to enable Sample Schemas and click **Next**.
The **Initialization Parameters** window appears.

Step 11 Under the Memory tab the default value is for a database instance with 4GB of memory. This can be set higher or lower as needed.

Note The amount of memory used should not be configured too high as this starves other database instances of memory.

Step 12 Under the Character Sets tab click the **Use Unicode** radio button and click **Next**.
The **Database Storage** window appears.

Step 13 Leave the default settings as they are and click **Next**.
The **Create Options** window appears.

Step 14 Check the Create Database check box and click **Finish**.

Step 15 Once a new database instance is created, you must temporarily change the ORACLE_SID environment variable (from Step 5) on your Unix system by running the command:

```
export ORACLE_SID=new_oracle_db_instance_sid.
```

This will change the SID so when you login using sqlplus, it will use the new instance and not the old one; you can then repeat the steps in Install Oracle Database, on page 11.

Once these steps are completed you can change the ORACLE_SID environment variable by sourcing the bash profile(assuming the old SID is in the bash profile) or by running the export command (Step 15) but changing the SID back to its original value.

CHAPTER 4

Install Microsoft SQL Server

This chapter provides information about installing and setting up Microsoft SQL.

Encrypted Database Not Supported

The IM and Presence Service does not support an encrypted database with Microsoft SQL Server, except in the following cases:

- The IM and Presence Service supports an encrypted compliance database for the Message Archiver feature. For 11.5(x) releases, this feature is supported as of 11.5(1)SU5. This feature is not supported for 12.0(x), but is supported for 12.5(1).

Install and Setup Microsoft SQL Server

Before you begin

- Read the security recommendations for the Microsoft SQL database in the About Security Recommendations section.
- For information on supported versions, see External Database Setup Requirements.
- To install the MS SQL Server, refer to your Microsoft documentation.

Note In compliance with XMPP specifications, the IM and Presence Service node uses UTF8 character encoding. This allows the node to operate using many languages simultaneously and to display special language characters correctly in the client interface. If you want to use Microsoft SQL with the node, you must configure it to support UTF8.

Connect to the MS SQL Server using **Microsoft SQL Server Management Studio**.

Create a New Microsoft SQL Server Database

Use this procedure to create a new Microsoft SQL Server database.

Procedure

Step 1 Enable SQL server and Windows authentication:

a) In the left navigation pane, right-click the name of the Microsoft SQL Server, then click **properties**.

b) Click **Enable SQL Server and Windows Authentication mode**.

Step 2 In the left navigation pane, right-click **Databases** and click **New Database**.

Step 3 Enter an appropriate name in the **Database name** field.

Step 4 Click **OK**. The new name appears in the left navigation pane nested under databases.

Configure MSSQL Named Instance

Microsoft SQL Server Browser service is responsible for listening on UDP port 1433 for incoming connections to a named instance. The SQL Server Browser service responds to the client with a dynamically assigned TCP port number, which is used for that session connection to the named instance.

IM and Presence does not support dynamic port allocation, so you need to configure the Microsoft SQL Server instance to use a static TCP port.

Use this procedure to statically assign a listening port for a named instance:

Procedure

Step 1 Log in to the Microsoft server where SQL Server is installed.

Step 2 Select **Start > Microsoft SQL Server > SQL Server Configuration.**

Step 3 In the SQL Server Configuration Manager, select **SQL Server Network Configuration > Protocols for <named_Instance_name>**, and then select the TCP\IP protocol name.

Step 4 In the TCP\IP properties for the named instance, select the **IP Address** tab. The configuration will have several IP configuration sections, such as IP1, IP2, IP3, IP4, IP5, IP6 and IPALL.

Step 5 Perform the following steps for each of the above referenced IP configuration sections:

a) Remove any configuration in the **<TCP Dynamic Ports>** field.

b) Choose a TCP port you want to use for the named instance and update the **TCP Port** field with the chosen port.

c) Add a firewall rule for the SQL named instance.

Note When configuring the external database in IM and Presence, make sure to update the SQL TCP port to the value that is defined in the previous steps.

Create a new Login and Database User

Use this procedure to create a new login and Microsoft SQL database user.

Procedure

Step 1 In the left navigation pane, right-click **Security** > **Login** and click **New Login**.

Step 2 Enter an appropriate name in the **Login name** field.

Step 3 Check the **SQL Server authentication** check box.

Step 4 Enter a new password in the **Password** field and confirm the password in the **Confirm password** field.

Step 5 Check the **Enforce password policy** check box.

Note Ensure that the **Enforce password expiration policy** is not checked. This password is used by IM and Presence Service to connect to the database and must not expire.

Step 6 Choose the database you want to apply this new user to from the **Default database** drop-down list.

Step 7 In the left navigation pane of the **Login - New** window, click **User Mapping**.

Step 8 Under the **Users mapped to this login** list, check the database to which you want to add this user.

Step 9 Click **User Mapping**, in the **Map** column of the **Users mapped to this pane** pane, check the check box of the database you have already created.

Step 10 In **Server Roles**, ensure that only the **public** role check box is checked.

Step 11 Click **OK**. In **Security** > **Logins**, the new user is created.

Grant Database User Owner Privileges

Use this procedure to grant ownership of a Microsoft SQL database to a database user.

Procedure

Step 1 In the left navigation pane click **Databases**, then click on the name of the database that you have created and click **Security** > **Users**.

Step 2 Right-click on the name of the database user to who you want to add owner privileges, then click **Properties**.

Step 3 In the Database User pane, click **Membership**.

Step 4 In the **Role Members** list, check the **db_owner** check box.

Step 5 Click **OK**.

[Optional] Database User Access Restrictions

Use this procedure if you want to remove the database user as the database owner and apply further optional restrictions to the database user on the Microsoft SQL Server database.

⚠

Caution If during an IM and Presence Service upgrade, there is a database schema upgrade, then the database user must have owner privileges for the database.

Before you begin

Ensure that you carry out the procedures in the Configure IM and Presence Service for External Database, on page 23 chapter.

Procedure

Step 1 Create a new database role for executing stored procedures:

a) In the left navigation pane click **Databases**, then click the name of the database to which you want to add new database roles.
b) Right-click **Roles**, and click **New Database Role**.
c) In the **Database Role** window, click **General**.
d) Enter an appropriate name in the **Role name** field.
e) Click **Securables**, then click **Search** to open the **Add Objects** window.
f) Choose the **Specific Objects** radio button, and click **OK**.
g) Click **Object Types** to open the **Select Object Types** window.
h) In the **Select Object Types** window, check the **Stored procedures** check box and click **OK**. Stored procedures is then added to the **Select these object types** pane.
i) Click **Browse**.
j) In the **Browse for Objects** window, check the following check boxes:

- [dbo][jabber_store_presence]
- [dbo][ud_register]
- [dbo][ps_get_affiliation]
- [dbo][tc_add_message_clear_old]
- [dbo][wlc_waitlist_update]

k) Click **OK**. The new names appear in the **Enter the object names to select** pane.
l) On the **Select Objects** window, click **OK**.
m) From the **Database Role** window, click the first entry in the list of objects in the **Securables** list.
n) In the **Explicit** list, check the **Grant** check box for the **Execute** permission.
o) Repeat step 13 and 14 for all objects in the **Securables** list.
p) Click **OK**.

A new database role is created in **Security** > **Roles** > **Database Roles**.

Step 2 To update the database user's database role membership:

a) Under **Security** > **Users**, right-click on the database user you have created, then click **Properties**.
b) In the **Database User** window, click **Membership** in the left navigation pane.
c) In the **Role Members pane**, uncheck the **db_owner** check box.

d) Check the check boxes for **db_datareader**, **db_datawriter**, and the database role which you created in step 1.

Step 3 Click **OK**.

Default Listener Port Setup for Microsoft SQL Server

Assign a TCP/IP port number to the SQL Server Database Engine as the default listener port.

Procedure

Step 1 From the SQL Server Configuration Manager, click **SQL Server Network Configuration** > **Protocols** > **TCP/IP** in the **Console**.

Step 2 In the **TCP/IP Properties** dialog box, on the IP Addresses tab, right-click on the IP address that you want to configure and then click **Properties**.

Step 3 Check the **TCP Dynamic Ports** dialog box, if it contains the value 0, delete the 0. This prevents the Database Engine from listening on dynamic ports.

Step 4 In the **IPn Properties** pane, type the port number you want this IP address to listen on in the **TCP Port** pane.

Step 5 Click **OK**.

Step 6 Click **SQL Server Services** in the **Console** pane.

Step 7 In the **Details** pane, right-click **SQL Server** (instance name) and then click **Restart**, to stop and restart the Microsoft SQL Server.

Database Migration Required for Upgrades with Microsoft SQL Server

If you have Microsoft SQL Server deployed as an external database with the IM and Presence Service and you are upgrading from 11.5(1), 11.5(1)SU1, or 11.5(1)SU2, you must create a new SQL Server database and migrate to the new database. This is required due to enhanced data type support in this release. If you don't migrate your database, schema verification failure will occur on the existing SQL Server database and services that rely on the external database, such as persistent chat, will not start.

After you upgrade your IM and Presence Service, use this procedure to create a new SQL Server database and migrate data to the new database.

Note This migration is not required for Oracle or PostgreSQL external databases.

Before You Begin

The database migration is dependent on the `MSSQL_migrate_script.sql` script. Contact Cisco TAC to obtain a copy.

Table 3:

Step	Task
Step 1	Create a snapshot of your external Microsoft SQL Server database.
Step 2	Create a new (empty) SQL Server database. For details, see the following chapters in the Database Setup Guide for the IM and Presence Service: **1.** "Microsoft SQL Installation and Setup"—See this chapter for details on how to create your new SQL server database on your upgraded IM and Presence Service. **2.** "IM and Presence Service External Database Setup"—After your new database is created, refer to this chapter to add the database as an external database in the IM and Presence Service.
Step 3	Run the System Troubleshooter to confirm that there are no errors with the new database. **1.** From Cisco Unified CM IM and Presence Administration, choose **Diagnostics** > **System Troubleshooter**. **2.** Verify that no errors appear in the **External Database Troubleshooter** section.
Step 4	Restart the Cisco XCP Router on all IM and Presence Service cluster nodes: **1.** From Cisco Unified IM and Presence Serviceability, choose **Tools** > **Control Center - Network Services**. **2.** From the **Server** menu, select an IM and Presence Service node and click **Go**. **3.** Under **IM and Presence Services**, select **Cisco XCP Router**, and click **Restart**.
Step 5	Turn off services that depend on the external database: **1.** From Cisco Unified IM and Presence Serviceability, choose **Tools** > **Control Center - Feature Services**. **2.** From the **Server** menu, select an IM and Presence node and click **Go**. **3.** Under **IM and Presence Services**, select the following services:. Cisco XCP Text Conference Manager Cisco XCP File Transfer Manager Cisco XCP Message Archiver **4.** Click **Stop**.
Step 6	Run the following script to migrate data from the old database to the new database `MSSQL_migrate_script.sql`. **Note** Contact Cisco TAC to obtain a copy of this script

Step	Task
Step 7	Run the System Troubleshooter to confirm that there are no errors with the new database. **1.** From Cisco Unified CM IM and Presence Administration, choose **Diagnostics** > **System Troubleshooter**. **2.** Verify that no errors appear in the **External Database Troubleshooter** section.
Step 8	Start the services that you stopped previously. **1.** From Cisco Unified IM and Presence Serviceability, choose **Tools** > **Control Center - Feature Services**. **2.** From the **Server** menu, select an IM and Presence node and click **Go**. **3.** Under **IM and Presence Services**, select the following services: Cisco XCP Text Conference Manager Cisco XCP File Transfer Manager Cisco XCP Message Archiver **4.** Click **Start**.
Step 9	Confirm that the external database is running and that all chat rooms are visible from a Cisco Jabber client. Delete the old database only after you're confident that the new database is working.

Upgrade Database Schema from IM and Presence Release 11.5(1) and Above

If you have Microsoft SQL database deployed as an external database with the IM and Presence Service, choose either of the following scenarios to upgrade the database schema.

Table 4: MSSQL Database Schema Upgrade Scenarios

Scenario	Procedure
Upgrade from IM and Presence Service 11.5(1), 11.5(1)SU1, or 11.5(1)SU2 release	For more information on how to upgrade your MSSQL database, see the 'Database Migration Required for Upgrades with Microsoft SQL Server' section in the Database Setup Guide for the IM and Presence Service. This makes the necessary changes to the column types from TEXT to nvarchar(MAX).

Scenario	Procedure
Upgrade from IM and Presence Service 11.5(1)SU3 or later	The MSSQL database connected to the IM and Presence Service Server is upgraded automatically during IM and Presence Service upgrade. This makes the necessary changes to the column types from nvarchar(4000) to nvarchar(MAX). **Note** If you want to trigger an upgrade manually for any reason, such as to connect to an older database with column type as nvarchar(4000), the following actions trigger and upgrade the database by changing the column type to nvarchar(MAX): • Restarting Cisco XCP Config Manager followed by restarting Cisco XCP Router service; or • During schema verification of the external database—when you assign the database to Text Conferencing (TC), Message Archiver (MA) or Managed File transfer (MFT) services, and reload the **External Database Settings** page. (From the Cisco Unified CM IM and Presence Administration user interface, choose **Messaging** > **External Server Setup** > **External Databases**, and then find and select the database to load the **External Database Settings** page.)

CHAPTER 5

Configure IM and Presence Service for External Database

This chapter provides information about configuring the IM and Presence Service for the external database connection.

About External Database Assignment

External Database and Node Assignment

When you configure an external database entry on the IM and Presence Service, you assign the external database to a node, or nodes, in your cluster as follows:

- Message Archiver (compliance) — You require at least one external database per cluster. Depending on your deployment requirements, you can also configure a unique external database per node.

- Persistent Group Chat — You require a unique external database per node. Configure and assign a unique external database for each node in your cluster.

- Managed File Transfer — You require one unique logical external database instance for each IM and Presence Service node, in an IM and Presence Service cluster/sub-cluster that has the Cisco XCP File Transfer Manager service activated.

- If you deploy the persistent group chat, message archiver, and managed file transfer features on an IM and Presence Service node, you can assign the same external database to all or any combination of the features.

For more information see:

- Message Archiver — *Instant Messaging Compliance for IM and Presence Service on Cisco Unified Communications Manager*.

- Persistent Group Chat — *Configuration and Administration of IM and Presence Service on Cisco Unified Communications Manager.*
- Managed File Transfer — *Configuration and Administration of IM and Presence Service on Cisco Unified Communications Manager*.

Related Topics

Set Up External Database Entry on IM and Presence Service, on page 24
External Database Connection, on page 24

External Database Connection

IM and Presence Service does not establish a connection to the external database when you configure an external database entry. The external database has not created the database schema at this point. It is only when you assign an external database entry to a node that IM and Presence Service establishes an ODBC (Open Database Connectivity) connection with the external database. Once IM and Presence Service establishes a connection, the external database creates the database tables for the IM and Presence Service features.

Once you assign an external database entry to a node, you can validate the connection using the System Troubleshooter in the **Cisco Unified CM IM and Presence Service Administration** user interface.

Related Topics

Set Up External Database Entry on IM and Presence Service, on page 24
Verify External Database Connection Status on IM and Presence Service

Set Up External Database Entry on IM and Presence Service

Perform this configuration on the IM and Presence Service database publisher node of your cluster.

Caution If your IM and Presence Service node connects to an external database server using IPv6, ensure that the enterprise parameter is configured for IPv6 and that Eth0 is set for IPv6 on each node in the deployment; otherwise, the connection to the external database server fails. The Message Archiver and Cisco XCP Text Conference Manager are unable to connect to the external database and fail. For information about configuring IPv6 on IM and Presence Service, see *Configuration and Administration of IM and Presence Service on Cisco Unified Communications Manager*.

Before you begin

- Install and configure the external database.
- Obtain the hostname or IP address of the external database.
- If using Oracle, retrieve the tablespace value. To determine the tablespace available for your Oracle database, execute the following query as sysdba:

```
SELECT DEFAULT_TABLESPACE FROM DBA_USERS WHERE USERNAME = 'USER_NAME';
```

Note The user name must be capitalized and in single quotes (a string literal) for this command to succeed, even if you defined the user with lowercase characters.

Procedure

Step 1 Log in to the **Cisco Unified CM IM and Presence Administration** user interface. Choose **Messaging** > **External Server Setup** > **External Databases**.

Step 2 Click **Add New**.

Step 3 Enter the name of the database that you defined at external database installation, for example, **`tcmadb`**.

Step 4 Choose the database type from the drop-down list, Postgres, Oracle, or Microsoft SQL Server.

Step 5 If you chose Oracle as the database type, enter the tablespace value.

Step 6 Enter the username for the database user (owner) that you defined at external database installation, for example, **`tcuser`**.

Step 7 Enter and confirm the password for the database user, for example, **`mypassword`**.

Note The password length for external database should be less or equal to 30 characters long.

Step 8 Enter the hostname or IP address for the external database.

Step 9 Enter a port number for the external database.

The default port numbers for Postgres (5432), Oracle (1521), Oracle with SSL enabled (2484), and Microsoft SQL Server (1433) are prepopulated in the **Port Number** field. You can choose to enter a different port number if required.

Step 10 If you chose Oracle or Microsoft SQL Server as the Database Type the **Enable SSL** check box becomes active. Check the check box to enable SSL.

Note

- If you chose Microsoft SQL Server as the Database Type, the **Certificate Name** drop-down list remains inactive because all certificates in the cup-xmpp-trust list are used to verify the certificate sent from the Microsoft SQL Server
- If you chose Microsoft SQL Server as the Database Type, the hostname must match the entry in the **Common Name** field of the certificate presented by the SQL Server.

If you chose Oracle as the Database Type, the **Certificate Name** drop-down list becomes active. Choose a certificate from the drop-down list.

Note

- When the Enable SSL check box or the Certificate drop-down field is modified, a notification to restart the corresponding service assigned to the external database is sent. A message concerning either Cisco XCP Message Archiver or Cisco XCP Text Conference Manager will be generated.
- The certificate you need to enable SSL must be uploaded to the cup-xmpp-trust store. You must upload this certificate before you enable SSL.
- Once the certificate is uploaded to the cup-xmpp-trust store, you must wait 15 minutes for the certificate to propagate to all the nodes of the IM and Presence Service cluster. If you do not wait, the SSL connection on nodes where the certificate has not propagated fails.
- If the certificate is missing or has been deleted from the cup-xmpp-trust store, an alarm XCPExternalDatabaseCertificateNotFound is raised in the Cisco Unified Communications Manager Real Time Monitoring Tool (RTMT).

  Note No alarm is raised if the external database type chosen is Microsoft SQL Server.

- The following ciphers have been tested with Microsoft SQL Server:
  - TLS_RSA_WITH_AES_128_CBC_SHA256
  - TLS_RSA_WITH_AES_128_CBC_SHA
  - TLS_RSA_WITH_AES_256_CBC_SHA256

Step 11 Click **Save**.

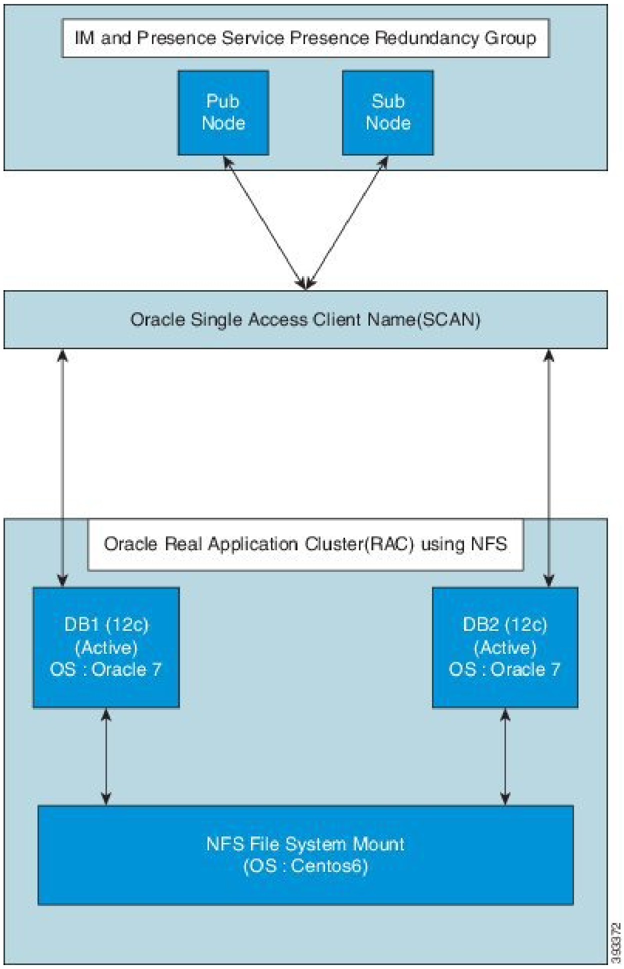
Related Topics

Verify External Database Connection, on page 29

External Database for Persistent Chat High Availability

For information on supported versions, refer to the External Database Setup Requirements section of the *Database Setup Guide for IM and Presence Service*.

Figure 1: Oracle High Availability Setup


IM and Presence Service Presence Redundancy Group
Pub Node
Sub Node
Oracle Single Access Client Name(SCAN)
Oracle Real Application Cluster(RAC) using NFS
DB1 (12c) (Active) OS : Oracle 7
DB2 (12c) (Active) OS : Oracle 7
NFS File System Mount (OS : Centos6)
393372

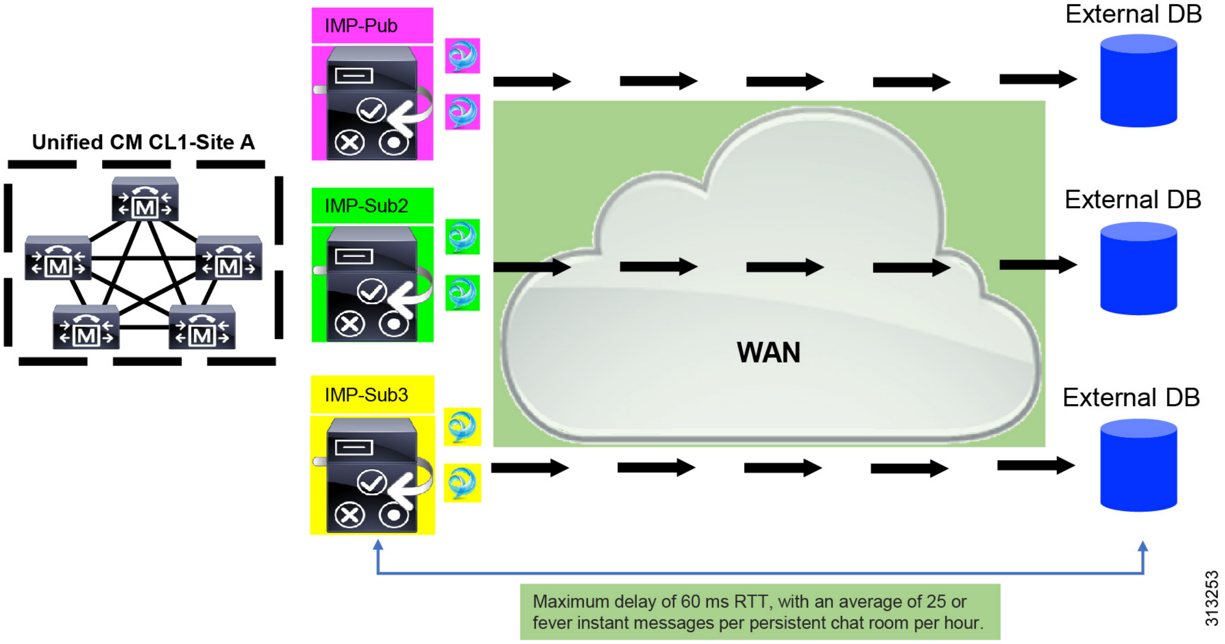
Message Archiver Network Latency

https://ciscoenterprise.acrolinx.cloud for the Acrolinx URL

Cisco IM and Presence Service is enabled for persistent chat, message archiving, or compliance logging. We recommend to place the external database servers on the same side of the WAN as the associated Cisco IM and Presence Service subclusters to keep a low network latency.

In case the external database cannot be located on the same side of the WAN as its associated Cisco IM and Presence Service subcluster, the following considerations apply with respect to deployment profile and latency.

Figure 2: Message Archiver with External Database Across the WAN

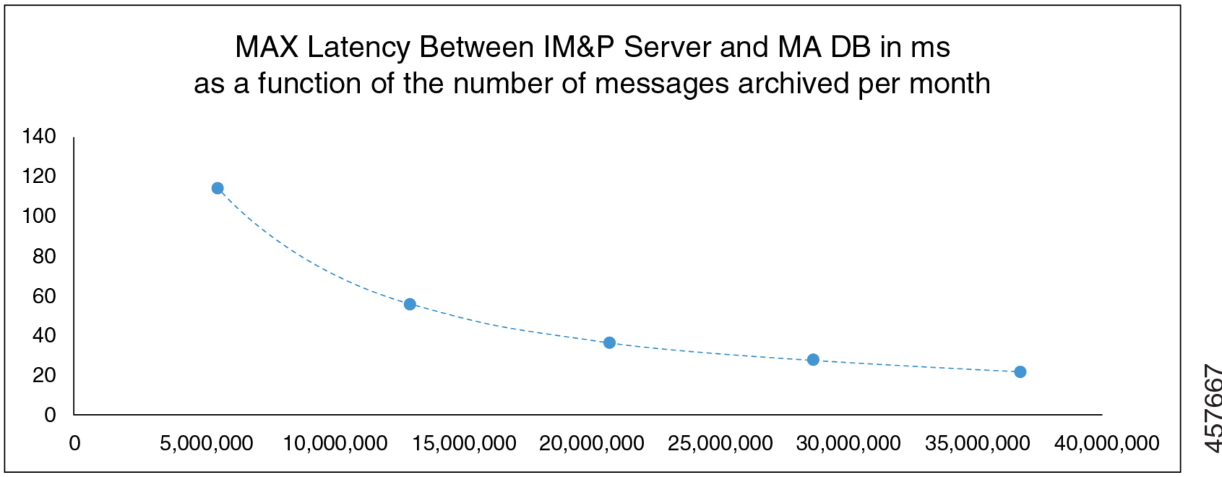

IMP-Pub
IMP-Sub2
IMP-Sub3
Unified CM CL1-Site A
WAN
External DB
External DB
External DB
Maximum delay of 60 ms RTT, with an average of 25 or fever instant messages per persistent chat room per hour.
313253

The Message Archiver feature is latency dependent. When modeling acceptable network latency due to round-trip time (RTT), the following parameters apply:

- Number of users per Cisco IM and Presence Service
- Average number of messages per time interval
- Number of active chat rooms
- Average number of users per active chat room
- Average number of messages per active chat room per time interval
- Average number of logged in devices per user

As network latency increases, the number of users, devices, and/or messages per user must decrease.

Figure 3: Max Latency/RTT plot



Calculating the Max Latency/RTT

Use the guideline we provide here to calculate the Max Latency/RTT.

Guideline

The Max Latency/RTT time (in ms) for the given number of monthly archived messages is calculated using the following formula:

Max RTT=$8*10^7*TMM^{-0.867}$

Where TMM represents the total number of archived messages per month.

Verify External Database Connection

If you make a configuration change in the `install_dir/data/pg_hba.conf` file or the `install_dir/data/postgresql.conf` file after you assign the external database, perform these steps:

Procedure

Step 1 Unassign and reassign the external database to the IM and Presence Service node.

Step 2 Restart the Cisco XCP Router service. Log in to the **Cisco Unified IM and Presence Serviceability** user interface. Choose **Tools** > **Control Center - Network Services** to restart this service.

Related Topics

Install PostgreSQL, on page 7
Install Oracle, on page 11
Install Microsoft SQL Server, on page 15

Verify External Database Connection Status on IM and Presence Service

IM and Presence Service provides the following status information on an external database:

- Database reachability — Verifies that the IM and Presence Service can ping an external database.
- Database connectivity — Verifies that the IM and Presence Service has successfully established an Open Database Connectivity (ODBC) connection with the external database.
- Database schema verification — Verifies that the external database schema is valid.

Caution If your IM and Presence Service node connects to an external database server using IPv6, ensure that the enterprise parameter is configured for IPv6 and that Eth0 is set for IPv6 on each node in the deployment; otherwise, the connection to the external database server fails. The message archiver (compliance) and Cisco XCP Text Conference Manager is unable to connect to the external database and fails. For information about configuring IPv6 on IM and Presence Service, see *Configuration and Administration of IM and Presence Service on Cisco Unified Communications Manager*.

Procedure

Step 1 Log in to the **Cisco Unified CM IM and Presence Administration** user interface. Choose **Messaging** > **External Server Setup** > **External Databases**.

Step 2 Click **Find**.

Step 3 Choose the external database entry that you want to view.

Step 4 Verify that there are check marks beside each of the result entries for the external database in the External Database Status section.

Step 5 In the **Cisco Unified CM IM and Presence Administration** user interface, choose **Diagnostics** > **System Troubleshooter**.

Step 6 Verify that there are check marks beside the status of each of the external database connection entries in the External Database Troubleshooter section.

Troubleshooting Tips

- The IM and Presence Service generates an alarm if it loses ODBC to an external database.
- Changing the password of the external database user when the IM and Presence Service is already connected to the external database will not affect the existing connections. When the Cisco XCP Router service is restarted, it destroys the existing connections and the new password is used for creating new connections.

  If the user password is changed only on the external database, the configured IM and Presence features relying on the external database continues to work using the existing connections established with the old user password. However, this transient phase shall be kept to a minimum and at the earliest ensure to update the password on the IM and Presence Service too.

- You can also verify the status of the Postgres database connection using the **psql** command. You must sign in to the Linux shell from a remote support account to run this command; it is not accessible through the administrator CLI. Run the following command after you install the Postgres database, but before you assign the database to an IM and Presence Service node.

  Important For `psql` to run, you must first set an environment variable by entering:

  ```
  $export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/xcp/lib
  ```

  Enter:

  ```
  $sudo -u xcpuser /usr/local/xcp/bin/psql -U db_user -h db_server db_name
  ```

  For example:

  ```
  $sudo –u xcpuser /usr/local/xcp/bin/psql –U postgres -h node1 tcmadb
  ```

- You can verify the status of the Oracle database connection by executing the following commands from the root:

  ```
  export ORACLE_HOME=/usr/lib/oracle/client_1/

  export PATH="$ORACLE_HOME/bin:$PATH"

  export LD_LIBRARY_PATH="$ORACLE_HOME/lib:$LD_LIBRARY_PATH"

  sqlplus username/password@dsn
  ```

  The *dsn* value can be obtained from the `$ORACLE_HOME/network/admin/tnsnames.ora` file.

- You can verify the status of the Microsoft SQL database connection by executing the following commands from the root:

  ```
  $sudo -u xcpuser TDSVER=7.3 /usr/local/xcp/bin/tsql -H mssql_server_hostname -p portnumber
  -U username -D databasename
  ```

- If you configure the message archiver (compliance) feature, and the Cisco XCP Message Archiver service fails to start, or you configure the persistent group chat feature and the Cisco Text Conference Manager service fails to start, check the External Database Troubleshooter section of the **System Configuration Troubleshooter** window.

  - If it shows that the status of the external database connection is not **OK**, verify that you provided the correct connection details and that there are no network issues between the IM and Presence Service node and the external database host.

  - If the status of the external database connection is **OK**, but the schema verification status is not, unassign the external database from the node, and reassign it again.

- Once the certificate is uploaded to the cup-xmpp-trust store, you must wait 15 minutes for the certificate to propagate to all the nodes of the IM and Presence Service cluster. If you do not wait, the SSL connection on nodes where the certificate has not propagated fails.
- If the certificate is missing or has been deleted from the cup-xmpp-trust store, an alarm 'XCPExternalDatabaseCertificateNotFound' is raised in the Cisco Unified Communications Manager `Real Time Monitoring Tool` (RTMT).

Note No alarm is raised if the external database type chosen is Microsoft SQL Server.

CHAPTER 6

Database Tables

This chapter provides information about the external database tables that are created in your schema to support the IM and Presence Service node.

Note By default, the IM and Presence Service generates 27 tables in the external database but at present it only uses the tables described in this module.

Note If you need to modify any data in the external database, ensure that you restart the Cisco XCP Text Conference Manager service after you have made those changes.

AFT_LOG Table

The AFT_LOG table, contains information about file transfers that occur when using the Cisco Unified Communications Manager IM and Presence Service managed file transfer feature.

Indexes: "aft_log_pkey" PRIMARY KEY, btree (aft_index)

Column Name	Postgres Datatype	Oracle Datatype	Microsoft SQL Datatype	Not Null	Description
AFT_INDEX	BIGINT	NUMBER (19)	bigint	Yes	The sequence number that identifies the transaction.

Column Name	Postgres Datatype	Oracle Datatype	Microsoft SQL Datatype	Not Null	Description
JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar (3071)	Yes	The Jabber ID (JID) of the user who uploaded or downloaded a file. The contents of this column depend on the contents of the METHOD column. • When the METHOD column contains "POST," this is the JID of the user who uploaded the file. • When the METHOD column contains "GET," this is the JID of the user who downloaded the file.
TO_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar (3071)	Yes	The JID of the user, group chat, or persistent room that is the intended recipient of the file transfer.
METHOD	VARCHAR (63)	VARCHAR2 (63)	nvarchar (63)	Yes	This column can contain either POST, which indicates a user has uploaded a file, or GET, which indicates a user has downloaded a file.
FILENAME	VARCHAR (511)	VARCHAR2 (511)	nvarchar (511)	Yes	The resource name for the file that was uploaded or downloaded. The resource name identifies the file in HTTP requests. It is autogenerated by the IM and Presence Service.
REAL_FILENAME	VARCHAR (511)	VARCHAR2 (511)	nvarchar (511)	Yes	The actual name of the file that was uploaded by a user.
FILE_TYPE	VARCHAR (10)	VARCHAR2 (10)	nvarchar (10)	Yes	The file extension, for example jpg, txt, pptx, docx, and so on.
CHAT_TYPE	VARCHAR (10)	VARCHAR2 (10)	nvarchar (10)	Yes	"im" if the file was transferred during a one-to-one IM conversation. "groupchat" if the file was transferred during an ad hoc group chat conversation. "persistent" if the file was transferred to a persistent chat room.
FILE_SERVER	VARCHAR (511)	VARCHAR2 (511)	nvarchar (511)	Yes	The hostname or IP address of the file server where the file is stored.
FILE_PATH	VARCHAR (511)	VARCHAR2 (511)	nvarchar (511)	Yes	The absolute path to the file (including the file name) on the file server. The file name as stored on the repository is unique and is auto-generated by the IM and Presence Service.
FILESIZE	BIGINT	NUMBER (19)	bigint	Yes	The size of the file in bytes.
BYTES_ TRANSFERRED	BIGINT	NUMBER (19)	bigint	Yes	The number of bytes that were transferred. This number differs from FILESIZE, only when an error occurred during the transfer.
TIMESTAMPVALUE	TIMESTAMP	TIMESTAMP	timestamp	Yes	The date and time (UTC) the file was uploaded or downloaded.

Sample SQL Queries for the AFT_LOG Table

This section contains some sample SQL queries that you can run on the AFT_LOG table to extract specific information.

All Uploaded Files

The following SQL query returns records of all the files and screen captures that were uploaded using the manged file transfer feature:

```
SELECT file_path

FROM aft_log
```

```
WHERE method = 'Post';
```

All Files That Were Uploaded to a Specific Recipient

The following SQL query returns the records of all the files and screen captures that were uploaded to the user <userid> using the managed file transfer feature.

Note Records of downloaded files and screen captures do not contain any data in the *to_jid* field.

```
SELECT file_path

FROM aft_log

WHERE to_jid = '<userid>@<domain>';
```

All Files That Were Uploaded by a Specific Sender

The following SQL query returns the records of all the files and screen captures that were uploaded by the user <userid> using the managed file transfer feature.

```
SELECT file_path

FROM aft_log

WHERE jid  LIKE '<userid>@<domain>%' AND method = 'Post';
```

All Files That Were Downloaded by a Specific User

The following SQL query returns the records of all the files and screen captures that were downloaded by the user <userid> using the manged file transfer feature.

```
SELECT file_path

FROM aft_log

WHERE jid  LIKE '<userid>@<domain>%' AND method = 'Get';
```

All Files That Were Uploaded and Downloaded During IM Conversations

The following SQL query returns the records of all the files and screen captures that were uploaded and downloaded in IM conversations using the managed file transfer feature.

```
SELECT file_path

FROM aft_log

WHERE chat_type = 'im';
```

All Files That Were Uploaded by a Specific User After a Specific Time

The following SQL query returns the records of all the files and screen captures that were uploaded by the user <userid> after a specific time using the managed file transfer feature.

```
SELECT file_path

FROM aft_log
```

```
WHERE jid  LIKE '<userid>@<domain>%' AND method = 'Post' AND timestampvalue > '2014-12-18
11:58:39';
```

Sample Output for SQL Queries for the AFT_LOG Table

Sample output from any of these queries looks like this:

```
/opt/mftFileStore/node_1/files/im/20140811/15/file_name1

/opt/mftFileStore/node_1/files/im/20140811/15/file_name2

/opt/mftFileStore/node_1/files/im/20140811/15/file_name3

/opt/mftFileStore/node_1/files/im/20140811/15/file_name4

...

/opt/mftFileStore/node_1/files/im/20140811/15/file_name99

/opt/mftFileStore/node_1/files/im/20140811/15/file_name100
```

Using the Output to Clean Up the External File Server

You can use this output with the **rm** command to remove unwanted files from the external file server. For example, you can run the following commands on the external file server:

```
rm /opt/mftFileStore/node_1/files/im/20140811/15/file_name1

rm /opt/mftFileStore/node_1/files/im/20140811/15/file_name2

rm /opt/mftFileStore/node_1/files/im/20140811/15/file_name3
```

and so on.

TC_ROOMS Table

The TC_ROOMS table contains information for group chat rooms.

Column Name	Postgres Datatype	Oracle Datatype	Microsoft SQL Datatype	Not Null	Description
ROOM_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar (3071)	Yes	The ID of the room.
CREATOR_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar (3071)	Yes	The ID of the user who created the room.
SUBJECT	VARCHAR (255)	VARCHAR2 (255)	nvarchar (255)	Yes	The current subject for the room.
TYPE	VARCHAR (32)	VARCHAR2 (32)	nvarchar (32)	Yes	The constraint check_type. This value must be either “ad or “persistent”.

Column Name	Postgres Datatype	Oracle Datatype	Microsoft SQL Datatype	Not Null	Description
CONFIG	TEXT	CLOB	nvarchar (MAX)	Yes	The entire packet from the last time the room was co This information enables the room to be reconfigure the room is recreated (for example, at start-up).
SPACKET	TEXT	CLOB	nvarchar (MAX)	Yes	The entire packet from the last time the subject was s room. This information enables the room subject to be when the room is recreated.
START_MSG_ID	BIGINT	NUMBER (19)	bigint	Yes	A sequence number that is used to populate the MSC column in the TC_MSGARCHIVE table. Do not modify this value.
NEXT_MSG_ID	BIGINT	NUMBER (19)	bigint	Yes	A sequence number that is used to populate the MSC column in the TC_MSGARCHIVE table. Do not modify this value.

TC_USERS Table

The TC_USERS table contains roles and affiliations, alternate names, and other data associated with group chat room users.

Column Name	Postgres Datatype	Oracle Datatype	Microsoft SQL Datatype	Not Null	Description
ROOM_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar (3071)	Yes	The ID of the room.
REAL_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar (3071)	Yes	The ID of a user in the room. This value is tl user, rather than an alternate name.
ROLE	VARCHAR (32)	VARCHAR2 (32)	nvarchar (32)	Yes	The role of the user in the room. This value one of the following: “none”, “hidden”, “visi or “moderator”.
AFFILIATION	VARCHAR (32)	VARCHAR2 (32)	nvarchar (32)	Yes	The affiliation of the user in the room. This v to one of the following: “none”, “outcast”, “n or “owner”.
NICK_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar (3071)	Yes	The ID of the room, plus the alternate name format is room@tc-server/nick.
REASON	VARCHAR (255)	VARCHAR2 (255)	nvarchar (255)	Yes	The reason entered when the user's affiliation
INITIATOR_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar (3071)	Yes	The ID of the room in which the configurat occurred.

TC_MESSAGES Table

The TC_MESSAGES table contains messages that are sent in group chat rooms.

Column Name	Postgres Datatype	Oracle Datatype	Microsoft SQL Datatype	Not Null	Description
MSG_ID	BIGINT	NUMBER (19)	bigint	Yes	The ID of the message. The MSG_ID is a unique each message per chat room; it is not globally un
ROOM_JID	VARCHAR (3071)	VARCHAR (3071)	nvarchar (3071)	Yes	The ID of the room to which the message was se
STAMP	TIMESTAMP	TIMESTAMP	datetime	Yes	The date and time the message was sent.
MSG	TEXT	CLOB	nvarchar (MAX)	Yes	The entire message.

TC_TIMELOG Table

The TC_TIMELOG table contains the time that users enter and exit specific group chat rooms. This table may be used in conjunction with the other TC tables to recreate group chat conversations and to determine which users viewed the conversations.

Column Name	Postgres Datatype	Oracle Datatype	Microsoft SQL Datatype	Not Null	Description
REAL_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar (3071)	Yes	The ID of the user who is entering or leaving the r
NICK_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar (3071)	Yes	The ID of the room, plus the alternate name for the The format is room@tc-server/nick.
DIRECTION	VARCHAR (1)	VARCHAR2 (1)	nvarchar (1)	Yes	Indicates whether the user entered (E) or left (L) th Constrained to the values “E” and “L”.
STAMP	TIMESTAMP	TIMESTAMP	datetime	Yes	The date and time at which the user entered or left th UTC format from IMP server.

TC_MSGARCHIVE Table

The TC_MSGARCHIVE table stores messages and associated information for group chat rooms.

Note This table archives all messages if you turn on group chat on IM and Presence Service. Choose the option Archive all room messages on the **Cisco Unified CM IM and Presence Administration** user interface. Choose **Messaging** > **Conferencing and Persistent Chat**. See *Configuration and Administration of IM and Presence Service on Cisco Unified Communications Manager* for information on the group chat feature.

Column Name	Postgres Datatype	Oracle Datatype	Microsoft SQL Datatype	Not Null	Description
MSG_ID	BIGINT	NUMBER (19)	bigint	Yes	A unique identifier for the messag
TO_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar (3071)	Yes	The ID of the room that received t
FROM_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar (3071)	Yes	The ID of the user who sent the m
NICK_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar (3071)	Yes	The ID of the room, plus the alterna for example: room@conference.exmpl.com/nic
SENT_DATE	TIMESTAMP	TIMESTAMP	datetime	Yes	The date the message sent. UTC fo
MSG_TYPE	VARCHAR (1)	VARCHAR2 (1)	nvarchar (1)	Yes	The first character of the type attri The possible values are “c” (chat), (groupchat), “h” (headline), and “
BODY_LEN	INT	NUMBER (9)	int	Yes	The length in characters of the me
MESSAGE_LEN	INT	NUMBER (9)	int	Yes	The length in characters of the me subject and body.
BODY_STRING	VARCHAR (4000)	VARCHAR2 (4000)	nvarchar (4000)	No	The message body.
MESSAGE_ STRING	VARCHAR (4000)	VARCHAR2 (4000)	nvarchar (4000)	No	The entire raw packet.
BODY_TEXT	TEXT	CLOB	nvarchar (MAX)	No	If the message body exceeds 4000 in this field rather than the BODY
MESSAGE_TEXT	TEXT	CLOB	nvarchar (MAX)	No	If the entire raw packet exceeds 40 stored in this column rather than i MESSAGE_STRING column.
SUBJECT	VARCHAR (255)	VARCHAR2 (255)	nvarchar (255)	No	The current subject of the room.

JM Table

The JM table stores conversations and associated information for the message archiver component. The message archiver component provides the native compliance functionality on the IM and Presence Service.

Column Name	Postgres Datatype	Oracle Datatype	Microsoft SQL datatype	Not Null	Description
TO_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar (3071)	Yes	The Jabber ID (JID) of the user who is message being archived.
FROM_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar (3071)	Yes	The JID of the user who is receiving th being archived.
SENT_DATE	TIMESTAMP	TIMESTAMP	datetime	Yes	The date the message sent. UTC forma server.
SUBJECT	VARCHAR (128)	VARCHAR2 (128)	nvarchar (128)	No	The subject line of the message that is archived.
THREAD_ID	VARCHAR (128)	VARCHAR2 (128)	nvarchar (128)	No	The thread ID of the message that is bein When a message thread is initiated, IM provides the value and all related mess thread will use this value. These values unique and identify the group of associat messages.
MSG_TYPE	VARCHAR (1)	VARCHAR2 (1)	nvarchar (1)	Yes	The first character of the message's typ The possible values are: • “c” — chat • “n” — normal • “g” — groupchat • “h” — headline • “e” — error
DIRECTION	VARCHAR (1)	VARCHAR2 (1)	nvarchar (1)	Yes	Indicates whether the message is “O” - or “I” — incoming. If the message is se users on the same server, it is logged tw as outgoing and once as incoming.
BODY_LEN	INT	NUMBER (9)	int	Yes	The number of characters in the messa
MESSAGE_LEN	INT	NUMBER (9)	int	Yes	The number of characters in the messag the subject and the body.
BODY_STRING	VARCHAR (4000)	VARCHAR2 (4000)	nvarchar (4000)	No	The message body.

Column Name	Postgres Datatype	Oracle Datatype	Microsoft SQL datatype	Not Null	Description
MESSAGE_ STRING	VARCHAR (4000)	VARCHAR2 (4000)	nvarchar (4000)	No	The entire raw packet.
BODY_TEXT	TEXT	CLOB	nvarchar (MAX)	No	If the message body exceeds 4000 stored in this field rather than the BO field.
MESSAGE_TEXT	TEXT	TEXT	nvarchar (MAX)	No	If the entire raw packet exceeds 40 it is stored in this field rather than i MESSAGE_STRING field.
HISTORY_FLAG	VARCHAR (1)	VARCHAR2 (1)	nvarchar (1)	Yes	Used when room history messages participants (upon entering an existi allows you to distinguish between m received while actively participatin those received as part of a history p message type is flagged with HISTORY_FLAG="H" in the datab this column is set to "N."

Sample SQL Queries for the JM Table

This section contains some sample SQL queries that you can run on the JM table to extract specific information. The following queries select all columns from the table but you can be more selective about which information you want to include in your SQL queries.

All Instant Messages Sent by a Specific User

The following SQL query returns all instant messages sent by a specific user:

```
SELECT to_jid, sent_date, subject, thread_id, msg_type, direction, body_len, message_len,

body_string, message_string, body_text, message_text, history_flag

FROM jm

WHERE from_jid like 'bob@cisco.com%';
```

All Instant Messages Received by a Specific User

The following SQL query returns all instant messages received by a specific user:

```
SELECT from_jid, sent_date, subject, thread_id, msg_type, direction, body_len,

message_len, body_string, message_string, body_text, message_text, history_flag

FROM jm

WHERE to_jid like 'bob@cisco.com%';
```

All Instant Messages That Contain a Specific Word

The following SQL query returns all instant messages that contain a specific word:

```
SELECT to_jid, from_jid, sent_date, subject, thread_id, msg_type, direction, body_len,

message_len, body_string, message_string, body_text, message_text, history_flag

FROM jm

WHERE LOWER(body_string) like LOWER('%hello%');
```

All Instant Messages Conversations and Chat Rooms From a Specific Date

The following SQL query returns all instant messages, conversations and chat rooms from a specific date:

```
SELECT to_jid, from_jid, sent_date, subject, thread_id, msg_type, direction, body_len,

message_len, body_string, message_string, body_text, message_text, history_flag

FROM jm

WHERE CAST(sent_date AS Character(32)) like '2011-01-31%';
```

CHAPTER 7

Database Administration

External Database Cleanup Utility

The External Database Cleanup Utility makes it easy for administrators to manage external database growth, thereby ensuring that your system continues to perform at the optimum level. The utility lets you create jobs that monitor the external database on an ongoing basis, deleting old records automatically as they expire. This ensures that the external database has adequate space and that system performance is not impacted by unchecked database growth.

The External Database Cleanup Utility can be used to manage external database growth for the following IM and Presence Service features, each of which relies on the external database:

- Persistent Chat High Availability
- Managed File Transfer
- Message Archiver

Interactions

The following interactions apply:

- Records that are deleted from the database are deleted without archiving.
- You can run the Database Cleanup utility in offline mode.
- A persistent chat room configuration option is provided to override the cluster-wide setting for retention durations. This allows chat room owners to customize the settings within a controlled range. This is dependent on Jabber client changes to enable this menu option.

Stored Procedures

The External Database Cleanup Utility uses the following stored procedures to delete records

- `tc_msgarchive_auto_cleanup`

- `tc_timelog_auto_cleanup`
- `aft_log_auto_cleanup`
- `im_auto_cleanup`

Run the External Database Cleanup Utility

Use this procedure to run the External Database Cleanup Utility to delete expired records from the external database. You can run a manual cleanup to complete a one-time deletion of records from the database or you can configure the system to monitor and delete records from the external database automatically.

Procedure

Step 1 Log into Cisco Unified CM IM and Presence Administration on the database publisher node.

Step 2 Choose **Messaging** > **External Server Setup** > **External DataBase Jobs**.

Step 3 Click **Clear External DB**.

Step 4 Do one of the following:

- For manual cleanup of an external database that connects to the publisher node, select **SameCup Node**.
- For manual cleanup of an external database that connects to a subscriber node, select **Other CupNode** and then select the external database details.
- If you are configuring the system to monitor and clean the external database automatically, check the **Automatic Clean-up** radio button.

Note We recommend that you run a manual cleanup prior to setting up the automatic cleanup.

Step 5 Set the **Number of Days** that you want to go back for file deletion. For example, if you enter 90, the system deletes records that are older than 90 days.

Step 6 Click **Update Schema** to create the Indexes and stored procedures for the database.

Note You need to update the schema only the first time that you run the job.

Step 7 Set the **Number of Days** that you want to go back for file deletion. For example, if you enter **90**, the system deletes records that are older than 90 days.

Step 8 In the **Feature Tables** section, select each feature for which you want to clean records:

- **Text Conference (TC)**—Select this option to clean database tables for the Persistent Chat feature.
- **Message Archiver (MA)**—Select this option to clean database tables for the Message Archiver feature.
- **Managed File Transfer (MFT)**—Select this option to clean database tables for the Managed File Transfer feature

Step 9 Click **Submit Clean-up Job**.

Note If you have the **Automatic** option enabled, and you want to disable it, click the **Disable Automatic Clean-up Job** button.

What to do next

If you just ran a manual cleanup, repeat the procedure and select **Automatic Clean-up** to set the utility to monitor and delete records automatically.

Troubleshooting the External Database Cleanup Utility

If you run into issues with the external database utility:

- Make sure that the database publisher node can connect to the external database and that the database has been provisioned. You can verify this on the database publisher node by selecting **Messaging** > **External Server Setup** > **External Databases** > **External DB Configuration**.

- For PostgreSQL databases, make sure that the IM and Presence database publisher node has full access permission to all of the other configured external databases.

Logs for the External Database Cleanup Tool are available under admin logs: `/var/log/active/tomcat/logs/cupadmin/log4j/`

Merge External Databases

Use this procedure to merge external databases.

Note Microsoft SQL database is not supported for merging external databases.

Optional. If you have upgraded from a release prior to 11.5(1), and multiple external databases were used to manage redundancy, use the External Database Merge Tool to merge your external databases into a single database.

Example

If you have upgraded from a release prior to 11.5(1), and you had persistent chat configured with each persistent chat node connecting to a separate external database instance, use this procedure to merge the two databases in a subcluster into a single database that connects to both nodes.

Before you begin

- Ensure that the two source destination databases are assigned correctly to each IM and Presence Service node in the presence redundancy group. This verifies that both of their schemas are valid.

- Back up the tablespace of the destination database.

- Ensure that there is enough space in the destination database for the new merged databases.

- Ensure that the database users, created for the source and destination databases, have the permissions to run these commands:

  - `CREATE TABLE`

  - `CREATE PUBLIC DATABASE LINK`

- If your database users do not have these permissions, you can use these commands to grant them:

- PostgreSQL:

  `CREATE EXTENTION`—This creates the dblink and requires super user or dbowner privileges. After this, you EXECUTE privilege for dblink by running following:

  ```
  GRANT EXECUTE ON FUNCTION DBLINK_CONNECT(text) to <user>
  ```

  ```
  GRANT EXECUTE ON FUNCTION DBLINK_CONNECT(text,text) to <user>
  ```

- Oracle:

  ```
  GRANT CREATE TABLE TO <user_name>;
  ```

  ```
  GRANT CREATE PUBLIC DATABASE LINK TO <user_name>;
  ```

- If you are using a PostgreSQL external database, make sure that the following access is configured in the pg_hba.conf file:
  - The IM and Presence publisher node must have full access to each external database.
  - The external PostgreSQL database must have full access to each database instance. For example, if the external database is configured on 192.168.10.1 then each database instance must be configured in the pg_hba.conf file as `host dbName username 192.168.10.0/24 password`.

Procedure

Step 1 Sign in to **Cisco Unified CM IM and Presence Administration** on the IM and Presence Service publisher node.

Step 2 Stop the Cisco XCP Text Conference Service on the **System** > **Services** window for each IM and Presence Service node in the presence redundancy group.

Step 3 Click **Messaging** > **External Server Setup** > **External Database Jobs**.

Step 4 Click **Find** if you want to see the list of merge jobs. Choose **Add Merge Job** to add a new job.

Step 5 On the **Merging External Databases** window, enter the following details:

- Choose **Oracle** or **Postgres** from the **Database Type** drop-down list.
- Choose the IP address and hostname of the two source databases and the destination database that will contain the merged data.

If you chose Oracle as the **Database Type** enter the tablespace name and database name. If you chose Postgres as the **Database Type** you provide the database name.

`Example`

Select DB1 as external database currently in use which is mapped to the persistent chat room.

Select DB2 as destination database which is an external database on which data needs to be merged, DB2 database should not be empty and should have atleast one persistent chat room which meets the criteria MINVALUE(1).

After the merge, the data of source database DB1 and destination database DB2 will be saved in destination database DB2. Also, DB1 and DB2 will have unique data.

Step 6 In the **Feature Tables** pane, the Text Conference(TC) check-box is checked by default. For the current release, the other options are not available.

Step 7 Click **Validate Selected Tables**.

Note If the Cisco XCP Text Conference service has not been stopped you receive an error message. Once the service has been stopped, validation will complete.

Step 8 If there are no errors in the **Validation Details** pane, click **Merge Selected Tables**.

Step 9 When merging has completed successfully, the **Find And List External Database Jobs** window is loaded. Click Find to refresh the window and view the new job.

Click **Find** to refresh the window and view the new job.

Click the **ID** of the job if you want to view its details.

Step 10 Restart the Cisco XCP Router service.

Step 11 Start the Cisco XCP Text Conference Service on both IM and Presence Service nodes.

Step 12 You must reassign the newly merged external database (destination database) to the presence redundancy group

Migrate Persistent Chat Rooms from One External Database to Another

You can move all persistent chat rooms from an existing external database to a new database of the same type or different types without changing the IM and Presence nodes. This enables you to migrate all persistent chat rooms from one database to another, for example, from Oracle to Oracle, Oracle to MSSQL, MSSQL to PostgreSQL, and so on.

In this context, when a new database in connected to any of the IM and Presence services such as Persistent Chat, Message Archiver or Managed File Transfer, the service first checks whether the IM and Presence schema is present in the database. If the schema exists, it reuses the same. However, it creates a new schema only if the database does not have the required IM and Presence schema.

After the data migration, you can verify either through admin login into your application or through the backend of your newly configured external database by running the respective select statements for the tables.

Note There is no specific level of access required for database modification.

In the following procedures, migrating persistent chat rooms from Oracle to PostgreSQL has been considered and the tools used for import/export of data mentioned in this procedure are examples only. You can choose any tools for this purpose.

Before you begin

You need to set up and configure the new external database on the IM and Presence node. Refer to Set Up External Database Entry on IM and Presence Service, on page 24 for more details.

Procedure

Step 1 Export the persistent chat rooms from the existing external database on the IM and Presence node.

Step 2 Import the data to the new database.

Step 3 Assign the configured external database entry on the respective IM and Presence Node.

Step 4 Restart the following presence services, such as XCP Router, Text Conference Manager and Message Archiver.

Step 5 To verify After restarting the services, log in to the Cisco Jabber and verify whether the chat rooms are available.

Example Scenarios

The following procedures are included to provide more clarity on how to migrate persistent chat rooms between various external databases. We have considered migrating persistent chat rooms in Cisco Jabber between various database types such as Oracle, PostgreSQL and MSSQL. The tools used for import/export of data mentioned in these procedures are examples only. However, you can choose any tools for this purpose.

- Migrate Persistent Chat Rooms from Oracle to PostgreSQL, on page 48
- Migrate Persistent Chat Rooms from Oracle to MSSQL, on page 49
- Migrate Persistent Chat Rooms between Two Oracle Databases, on page 51
- Migrate Persistent Chat Rooms from MSSQL to PostgreSQL, on page 52
- Migrate Persistent Chat Rooms from MSSQL to Oracle, on page 54
- Migrate Persistent Chat Rooms between Two MSSQL Databases, on page 55

Migrate Persistent Chat Rooms from Oracle to PostgreSQL

The following procedure shows how to migrate persistent chat rooms created in Jabber which is currently connected to Oracle database to a newly created PostgreSQL database configured as an external database for the same IM and Presence node.

Before you begin

- Persistent chat rooms are available in Jabber.
- You have configured the new external database, PostgreSQL in this case, in your environment.
- Ensure that you have installed a database migration tools in your machine such as, **Oracle SQL Developer** to export data from Oracle and **Table Plus** to import data into PostgreSQL in this case.

Procedure

Step 1 Export the persistent chat rooms from your existing Oracle database. To do this:

- Open the Oracle SQL Developer tool and connect to your existing Oracle database by providing details such as database name, username, password, hostname, and port.
- After establishing the connection, run the following query to view the persistent chat rooms, which you have created in Jabber:

```
select * from tc_rooms;
```

- From the object tree view, right click on the table that you want to export.
- Select CSV as the export data format.
- Browse the destination folder.
- Click **Next** and **Finish**.

This exports the selected table data in an Excel file.

Step 2 Import the persistent chat rooms to the new PostgreSQL database. To do this:

- Open the Table Plus tool and connect to the new PostgreSQL database by providing details such as database name, username, password, hostname, and port.
- Import the Excel file that was previously exported from the Oracle database.

You can view the imported table name in the tree structure.

Step 3 Assign the configured external database entry, that is PostgreSQL to the IM and Presence node. For more information on how to assign the external database on the IM and Presence node, see Set Up External Database Entry on IM and Presence Service, on page 24.

Step 4 Restart the following services:

- XCP Router
- Text Conference Manager
- Message Archiver

Message Archiver configuration is not mandatory until and unless the persistent chat room users communicate their chat messages in the room.

Step 5 You can verify the data migration in either of the following ways:

- Log in to Jabber as an admin and verify if the chat rooms are present.
- Verify the migration in the target database by running select statements for the following tables:
  - Persistent chat tables - {tc_users, tc_rooms, tc_messages, tc_msgarchive and tc_timelog}
  - Message Archiver - {JM}
  - Managed File Transfer - {aft_log}

Migrate Persistent Chat Rooms from Oracle to MSSQL

The following procedure shows how to migrate persistent chat rooms created in Jabber which is currently connected to Oracle database to a newly created MSSQL database configured as an external database for the same IM and Presence node.

Before you begin

- Persistent chat rooms are available in Jabber.

- You have configured the new external database, MSSQL in this case, in your environment.
- Ensure that you have installed a database migration tools in your machine such as, **Oracle SQL Developer** to export data from Oracle and **Microsoft SQL Server Management Studio** to import data into MSSQL in this case.

Procedure

Step 1 Export the persistent chat rooms from your existing Oracle database. To do this:

- Open the Oracle SQL Developer tool and connect to your existing Oracle database by providing details such as database name, username, password, hostname, and port.
- After establishing the connection, run the following query to view the persistent chat rooms, which you have created in Jabber:

```
select * from tc_rooms;
```

- From the object tree view, right click on the table that you want to export.
- Select CSV as the export data format.
- Browse the destination folder.
- Click **Next** and **Finish**.

This exports the selected table data in an Excel file.

Step 2 Import the persistent chat rooms to the new PostgreSQL database. To do this:

- Open the Microsoft SQL server Management Studio tool and connect to the new MSSQL database by providing details such as database name, username, password, hostname, and port.
- Import the Excel file that was previously exported from the Oracle database.

You can view the imported table name in the tree structure.

Step 3 Assign the configured external database entry, that is MSSQL to the IM and Presence node. For more information on how to assign the external database on the IM and Presence node, see Set Up External Database Entry on IM and Presence Service, on page 24.

Step 4 Restart the following services:

- XCP Router
- Text Conference Manager
- Message Archiver

Message Archiver configuration is not mandatory until and unless the persistent chat room users communicate their chat messages in the room.

Step 5 You can verify the data migration in either of the following ways:

- Log in to Jabber as an admin and verify if the chat rooms are present.
- Verify the migration in the target database by running select statements for the following tables:

- Persistent chat tables - {tc_users, tc_rooms, tc_messages, tc_msgarchive and tc_timelog}
- Message Archiver - {JM}
- Managed File Transfer - {aft_log}

Migrate Persistent Chat Rooms between Two Oracle Databases

The following procedure shows how to migrate persistent chat rooms created in Jabber which is currently connected to Oracle database to a newly created instance of Oracle database configured as an external database for the same IM and Presence node.

Before you begin

- Persistent chat rooms are available in Jabber.
- You have configured the new external database, Oracle in this case, in your environment.
- Ensure that you have installed a database migration tools in your machine such as, **Oracle SQL Developer** to export and import data from Oracle.

Procedure

Step 1 Create a new external database.

Step 2 Stop Text Conference Manager service on all the nodes.

Step 3 Assign the configured external database entry, that is Oracle to the IM and Presence node. For more information on how to assign the external database on the IM and Presence node, see Set Up External Database Entry on IM and Presence Service, on page 24.

Step 4 Check the IM and Presence troubleshooting page and ensure the database schema verification is successful.

Step 5 Check the databases, ensure the tables, indexes, and stored procedures have been created.

If something is not created, run the `postDBTool_oracle.sql` script directly on the database.

The script can be found on IM and Presence server under `/usr/local/xcp/schemas/sql/postDBTool_oracle.sql`.

Step 6 Export the persistent chat rooms from your existing Oracle database. To do this:

- Open the Oracle SQL Developer tool and connect to your existing Oracle database by providing details such as database name, username, password, hostname, and port.
- After establishing the connection, run the following query to view the persistent chat rooms, which you have created in Jabber:

```
select * from tc_rooms;
```

- From the object tree view, right click on the table that you want to export.
- Select CSV as the export data format.
- Browse the destination folder.

- Click **Next** and **Finish**.

This exports the selected table data in an Excel file.

Step 7 Import the persistent chat rooms to the new Oracle database. To do this:

- Open the Oracle SQL Developer tool and connect to the new Oracle database by providing details such as database name, username, password, hostname, and port.
- Import the Excel file that was previously exported from the Oracle database.

You can view the imported table name in the tree structure.

Step 8 Restart the following services:

- XCP Router
- Text Conference Manager
- Message Archiver

Message Archiver configuration is not mandatory until and unless the persistent chat room users communicate their chat messages in the room.

Step 9 You can verify the data migration in either of the following ways:

- Log in to Jabber as an admin and verify if the chat rooms are present.
- Verify the migration in the target database by running select statements for the following tables:
  - Persistent chat tables - {tc_users, tc_rooms, tc_messages, tc_msgarchive and tc_timelog}
  - Message Archiver - {JM}
  - Managed File Transfer - {aft_log}

Migrate Persistent Chat Rooms from MSSQL to PostgreSQL

The following procedure shows how to migrate persistent chat rooms created in Jabber which is currently connected to MSSQL database to a newly created PostgreSQL database configured as an external database for the same IM and Presence node.

Before you begin

- Persistent chat rooms are available in Jabber.
- You have configured the new external database, PostgreSQL in this case, in your environment.
- Ensure that you have installed a database migration tool in your machine such as, **Oracle SQL Developer** to export data from MSSQL and **Microsoft SQL Server Management Studio** to import data into PostgreSQL in this case.

Procedure

Step 1 Export the persistent chat rooms from your existing MSSQL database. To do this:

- Open the Oracle SQL Developer tool and connect to your existing MSSQL database by providing details such as database name, username, password, hostname, and port.
- After establishing the connection, run the following query to view the persistent chat rooms, which you have created in Jabber:

  ```
  select * from tc_rooms;
  ```

- From the object tree view, right click on the table that you want to export.
- Select CSV as the export data format.
- Browse the destination folder.
- Click **Next** and **Finish**.

This exports the selected table data in an Excel file.

Step 2 Import the persistent chat rooms to the new PostgreSQL database. To do this:

- Open the Microsoft SQL server Management Studio tool and connect to the new PostgreSQL database by providing details such as database name, username, password, hostname, and port.
- Import the Excel file that was previously exported from the MSSQL database.

You can view the imported table name in the tree structure.

Step 3 Assign the configured external database entry, that is PostgreSQL to the IM and Presence node. For more information on how to assign the external database on the IM and Presence node, see Set Up External Database Entry on IM and Presence Service, on page 24.

Step 4 Restart the following services:

- XCP Router
- Text Conference Manager
- Message Archiver

Message Archiver configuration is not mandatory until and unless the persistent chat room users communicate their chat messages in the room.

Step 5 You can verify the data migration in either of the following ways:

- Log in to Jabber as an admin and verify if the chat rooms are present.
- Verify the migration in the target database by running select statements for the following tables:
  - Persistent chat tables - {tc_users, tc_rooms, tc_messages, tc_msgarchive and tc_timelog}
  - Message Archiver - {JM}

- Managed File Transfer - {aft_log}

Migrate Persistent Chat Rooms from MSSQL to Oracle

The following procedure shows how to migrate persistent chat rooms created in Jabber which is currently connected to MSSQL database to a newly created Oracle database configured as an external database for the same IM and Presence node.

Before you begin

- Persistent chat rooms are available in Jabber.
- You have configured the new external database, Oracle in this case, in your environment.
- Ensure that you have installed a database migration tools in your machine such as, **Microsoft SQL Server Management Studio** to export data from Oracle and **Oracle SQL Developer** to import data into PostgreSQL in this case.

Procedure

Step 1 Export the persistent chat rooms from your existing MSSQL database. To do this:

- Open the Microsoft SQL Server Management Studio tool and connect to your existing MSSQL database by providing details such as database name, username, password, hostname, and port.
- After establishing the connection, run the following query to view the persistent chat rooms, which you have created in Jabber:

  ```
  select * from tc_rooms;
  ```

- From the object tree view, right click on the table that you want to export.
- Select CSV as the export data format.
- Browse the destination folder.
- Click **Next** and **Finish**.

This exports the selected table data in an Excel file.

Step 2 Import the persistent chat rooms to the new Oracle database. To do this:

- Open the Oracle SQL Developer tool and connect to the new Oracle database by providing details such as database name, username, password, hostname, and port.
- Import the Excel file that was previously exported from the MSSQL database.

You can view the imported table name in the tree structure.

Step 3 Assign the configured external database entry, that is Oracle to the IM and Presence node. For more information on how to assign the external database on the IM and Presence node, see Set Up External Database Entry on IM and Presence Service, on page 24.

Step 4 Restart the following services:

- XCP Router
- Text Conference Manager
- Message Archiver

Message Archiver configuration is not mandatory until and unless the persistent chat room users communicate their chat messages in the room.

Step 5 You can verify the data migration in either of the following ways:

- Log in to Jabber as an admin and verify if the chat rooms are present.
- Verify the migration in the target database by running select statements for the following tables:
    - Persistent chat tables - {tc_users, tc_rooms, tc_messages, tc_msgarchive and tc_timelog}
    - Message Archiver - {JM}
    - Managed File Transfer - {aft_log}

Migrate Persistent Chat Rooms between Two MSSQL Databases

The following procedure shows how to migrate persistent chat rooms created in Jabber which is currently connected to MSSQL database to a newly created instance of MSSQL database configured as an external database for the same IM and Presence node.

Before you begin

- Persistent chat rooms are available in Jabber.
- You have configured the new external database, MSSQL in this case, in your environment.
- Ensure that you have installed a database migration tools in your machine such as, **Microsoft SQL Server Management Studio** to export and import data from MSSQL.

Procedure

Step 1 Export the persistent chat rooms from your existing MSSQL database. To do this:

- Open the Microsoft SQL Server Management Studio tool and connect to your existing MSSQL database by providing details such as database name, username, password, hostname, and port.
- After establishing the connection, run the following query to view the persistent chat rooms, which you have created in Jabber:

```
select * from tc_rooms;
```

- From the object tree view, right click on the table that you want to export.
- Select CSV as the export data format.
- Browse the destination folder.
- Click **Next** and **Finish**.

This exports the selected table data in an Excel file.

Step 2 Import the persistent chat rooms to the new MSSQL database. To do this:

- Open the Microsoft SQL Server Management Studio tool and connect to the new MSSQL database by providing details such as database name, username, password, hostname, and port.
- Import the Excel file that was previously exported from the MSSQL database.

You can view the imported table name in the tree structure.

Step 3 Assign the configured external database entry, that is MSSQL to the IM and Presence node. For more information on how to assign the external database on the IM and Presence node, see Set Up External Database Entry on IM and Presence Service, on page 24.

Step 4 Restart the following services:

- XCP Router
- Text Conference Manager
- Message Archiver

Message Archiver configuration is not mandatory until and unless the persistent chat room users communicate their chat messages in the room.

Step 5 You can verify the data migration in either of the following ways:

- Log in to Jabber as an admin and verify if the chat rooms are present.
- Verify the migration in the target database by running select statements for the following tables:
  - Persistent chat tables - {tc_users, tc_rooms, tc_messages, tc_msgarchive and tc_timelog}
  - Message Archiver - {JM}
  - Managed File Transfer - {aft_log}