## CONTENTS

### CHAPTER 1

**External Database Installation and Setup**  1  
- External Database Setup Requirements  1  
- Additional Documentation  2  
- External Database Setup Prerequisites  3  
- Hardware and Performance Recommendations  3  
- About Security Recommendations  3  
  - External Database Connection Security  3  
  - Maximum Connection Limit Setup  3  
  - Default Listener Port Setup  4  

### CHAPTER 2

**PostgreSQL Installation and Setup**  5  
- Install PostgreSQL Database  5  
- Set Up PostgreSQL Listening Port  7  
- User Access Restriction Recommendations  7  

### CHAPTER 3

**IM and Presence Service External Database Setup**  9  
- About External Database Assignment  9  
  - External Database and Node Assignment  9  
  - External Database Connection  10  
- Set Up External Database Entry on IM and Presence Service  10  
- Verify External Database Connection  11  
- Verify External Database Connection Status on IM and Presence Service  11  

### CHAPTER 4

**PostgreSQL Database Tables**  13  
- TC_ROOMS Table  13  
- TC_USERS Table  14  
- TC_MESSAGES Table  15  
- TC_TIMELOG Table  15  

Database Setup for IM and Presence Service on Cisco Unified Communications Manager, Release 9.0(1)
TC_MSGARCHIVE Table  15
JM Table  16
Sample SQL Queries for the JM Table  17
   All Instant Messages Sent by a Specific User  17
   All Instant Messages Received by a Specific User  18
   All Instant Messages That Contain a Specific Word  18
   All Instant Messages Conversations and Chat Rooms From a Specific Date  18
External Database Installation and Setup

This chapter provides information about how to configure an external database to store information synchronized from the Cisco Unified Communications Manager IM and Presence Service. The following IM and Presence Service features require an external database:

- Persistent Group Chat feature - IM and Presence Service supports two types of group chat, temporary (ad-hoc) chat and persistent chat. You do not require an external database for temporary chat to work. However, if you require persistent chat rooms on the IM and Presence Service, you must configure an external database.

- Instant Messaging Compliance - If you deploy the native Message Archiver (MA) component on the IM and Presence Service for compliance logging, you require an external database.

External Database Setup Requirements, page 1
Additional Documentation, page 2
External Database Setup Prerequisites, page 3
Hardware and Performance Recommendations, page 3
About Security Recommendations, page 3

External Database Setup Requirements

- **Hardware requirements**: A remote server on which you install the PostgreSQL databases.

- **Software requirements**:
  - IM and Presence Service, current release
  - PostgreSQL database, versions 8.3.x through 9.1.1
  - You can install the PostgreSQL database on either a Linux or a Windows operating system. See the PostgreSQL documentation for details on the supported operating systems and platform requirements.
  - IPv4 and IPv6 are supported.
You can also use Version 8.1.x of the PostgreSQL database, but the configuration of these versions may be different to the PostgreSQL database configuration described in this section. See the PostgreSQL documentation for details on how to configure these PostgreSQL database versions. If you use Version 8.1.x of the PostgreSQL database, the database configuration on IM and Presence Service is the same as described in this section.

Note

- External Database requirements for IM and Presence Service features:
  
The external database requirements differ depending on the features you want to deploy on IM and Presence Service:
  
  - Persistent Group Chat feature: You require one unique external database for each IM and Presence Service node in an IM and Presence Service cluster. Each node requires its own logical database, but nodes can share the same physical database installation.
  
  - Compliance feature: We highly recommend that you configure at least one external database for an IM and Presence Service cluster; however you may require more than one external database for a cluster depending on your server capacity.

Note

If you deploy both the Persistent Group Chat and Compliance features on an IM and Presence Service node, you can assign the same external database to both features.

Related Topics

- Hardware and Performance Recommendations, on page 3
- External Database Setup Prerequisites, on page 3
- PostgreSQL documentation

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 the external database, specifically the Persistent Group Chat and Compliance features. See the documentation specific to the feature you are deploying for the complete configuration:

- For information on configuring the Compliance feature on the IM and Presence Service, see Instant Messaging Compliance for IM and Presence Service on Cisco Unified Communications Manager

- For information on configuring the Persistent Group Chat feature on the IM and Presence Service, see Configuration and Administration of IM and Presence Service on Cisco Unified Communications Manager
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 *Installing Cisco Unified Communications Manager*.
- Configure the IM and Presence Service nodes as described in *Configuration and Administration of IM and Presence Service on Cisco Unified Communications Manager*.

Hardware and Performance Recommendations

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

- We recommend that you use similar hardware for both the external database and the IM and Presence Service nodes.
- We recommend that you maintain the external database according to the best practice guidelines described in the product documentation. If you do not properly maintain the external database, and you allow the external database to fill up, this causes performance problems in the IM and Presence Service cluster.

Related Topics

- PostgreSQL documentation

About Security Recommendations

External Database Connection Security

The IM and Presence Service does not provide a secure TLS/SSL connection to the external database. 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 Connection Limit Setup

**Note**

Use this section as a guideline if you wish to limit the number of connections to the database. This section is optional configuration.

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.
You must modify this guideline to suit your IM and Presence Service deployment and configuration. The guideline we provide here assumes that:

- You are running both the 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 IM and Presence Service administration interface.

To limit the number of 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 this guideline:

\[
\text{max_connections} = N \times 10 + \text{Additional Connections}
\]

- \( N \) is the number of nodes in your IM and Presence Service cluster.
- 10 is the default number of connections to the database on the IM and Presence Service, that is, five connections for the Compliance feature and five connections for the Persistent Group Chat feature. You can configure the number of database connections for the Persistent Group Chat feature on the IM and Presence Service Administration interface.
- Additional Connections represents any independent administration or database administrator (DBA) connections to the database server.

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 63.

**Related Topics**

- PostgreSQL documentation

### 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 PostgreSQL server. You can configure the default listening port in the `postgresql.conf` file located in the `<install_dir>/data` directory.

**Related Topics**

- Set Up PostgreSQL Listening Port, on page 7
PostgreSQL Installation and Setup

This chapter provides information about installing and setting up PostgreSQL.

- Install PostgreSQL Database, page 5
- Set Up PostgreSQL Listening Port, page 7
- User Access Restriction Recommendations, page 7

Install PostgreSQL Database

Before You Begin
Read the security recommendations for the PostgreSQL database in section About Security Recommendations, on page 3.

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 PostgreSQL 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 "tcmdb".
```
#CREATE DATABASE tcmdb 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:

<table>
<thead>
<tr>
<th>#</th>
<th>TYPE</th>
<th>DATABASE</th>
<th>USER</th>
<th>CIDR-ADDRESS</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>host</td>
<td>tcmadb</td>
<td>tcuser</td>
<td>10.89.99.0/24</td>
<td>password</td>
</tr>
<tr>
<td>2</td>
<td>host</td>
<td>dbinst</td>
<td>mauser</td>
<td>10.89.99.0/24</td>
<td>password</td>
</tr>
</tbody>
</table>

Step 5  Enter these commands to define passwords for the Postgres and 'tcuser' users:

```bash
# 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 IM and Presence Service.

Step 6  If you are running 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:

```bash
# 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, you must set the following values in the `postgresql.conf` file:

```bash
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:

```bash
> psql tcmadb -U postgres

# 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):

```bash
client_min_messages = log
log_duration = on
```
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)
```

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 Service 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:

<table>
<thead>
<tr>
<th># TYPE</th>
<th>DATABASE</th>
<th>USER</th>
<th>CIDR-ADDRESS</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>dbinst1</td>
<td>tcuser1</td>
<td>10.89.99.0/24</td>
<td>password</td>
</tr>
<tr>
<td>host</td>
<td>dbinst2</td>
<td>mauser1</td>
<td>10.89.99.0/24</td>
<td>password</td>
</tr>
</tbody>
</table>

Example of a less secure configuration:

<table>
<thead>
<tr>
<th># TYPE</th>
<th>DATABASE</th>
<th>USER</th>
<th>CIDR-ADDRESS</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>dbinst1</td>
<td>tcuser1</td>
<td>10.89.99.0/24</td>
<td>trust</td>
</tr>
<tr>
<td>host</td>
<td>dbinst2</td>
<td>all</td>
<td>10.89.99.0/24</td>
<td>password</td>
</tr>
</tbody>
</table>

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 5
PostgreSQL documentation
IM and Presence Service External Database Setup

This chapter provides information about the IM and Presence Service external database setup.

- About External Database Assignment, page 9
- Set Up External Database Entry on IM and Presence Service, page 10
- Verify External Database Connection, page 11
- Verify External Database Connection Status on IM and Presence Service, page 11

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:

- For the Compliance feature, you require at least one external database per cluster. Depending on your deployment requirements, you can also configure a separate external database per node.
- For the Persistent Group Chat feature, you require a unique external database per node. Configure and assign a unique external database for each node in your cluster.
- If you deploy both the Persistent Group Chat and Compliance features on an IM and Presence Service node, you can assign the same external database to both features.

For further information on configuring the Compliance feature on the IM and Presence Service, see *Instant Messaging Compliance for IM and Presence Service on Cisco Unified Communications Manager*.

For further information on configuring the Persistent Group Chat feature on the IM and Presence Service, see *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 10
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 IM and Presence Service administration.

Related Topics
- Set Up External Database Entry on IM and Presence Service, on page 10
- Verify External Database Connection Status on IM and Presence Service, on page 11

Set Up External Database Entry on IM and Presence Service

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

Before You Begin

- Install and configure the external database.
- Obtain the hostname or IP address of the external database.

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 Enter the username for the database user (owner) that you defined at external database installation, for example, tcmadb.

Step 5 Enter and confirm the password for the database user, for example, mypassword.

Step 6 Enter the hostname or IP address for the external database.

Step 7 Enter a port number for the external database.

The default port numbers for Postgres (5432) and Oracle (1521) are prepopulated in the Port Number field. You can choose to enter a different port number if required.

Step 8 Click Save.

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, you should verify the external database connection.
What to Do Next

- To assign an external database to a node for the Persistent Group Chat feature. Log in to the Cisco Unified CM IM and Presence Administration user interface. Choose Messaging > Group Chat and Persistent Chat. See Configuration and Administration of IM and Presence Service on Cisco Unified Communications Manager for further information on configuring the Persistent Group Chat feature on the IM and Presence Service.

- Assign an external database to a node for the Message Archiver feature. See Configuration and Administration of IM and Presence Service on Cisco Unified Communications Manager for further information on configuring the Message Archiver feature on the IM and Presence Service.

Related Topics

Verify External Database Connection, on page 11

Verify External Database Connection

If you make a configuration change in the <install_dir>/data/pg_hba.conf 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

PostgreSQL Installation and Setup, on page 5

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 IM and Presence Service can ping an external database.

- Database connectivity - verifies that IM and Presence Service has successfully established an ODBC connection with the external database.
Procedure

Step 1 Log into 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

• IM and Presence Service generates an alarm if it loses the ODBC connection to an external database.

• 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 this command after you install the Postgres database, but before you assign the database to a IM and Presence Service node. Enter:

`$sudo -u xcpuser /usr/local/xcp/bin/psql -U <database user> -h <database server> <database name>`

For example:

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

• If you configure the Compliance feature, and the Cisco XCP Message Archiver service fails to start, but the System Configuration Troubleshooter window shows that the status of the external database connection is OK, Cisco recommends that you unassign the external database from the node, and reassign it again.
CHAPTER 4

PostgreSQL Database Tables

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

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

- TC_ROOMS Table, page 13
- TC_USERS Table, page 14
- TC_MESSAGES Table, page 15
- TC_TIMELOG Table, page 15
- TC_MSGARCHIVE Table, page 15
- JM Table, page 16

**TC_ROOMS Table**

The TC_ROOMS table contains information for group chat rooms.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype</th>
<th>Not Null</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOM_JID</td>
<td>VARCHAR (3071)</td>
<td>Yes</td>
<td>The ID of the room.</td>
</tr>
<tr>
<td>CREATOR_JID</td>
<td>VARCHAR (3071)</td>
<td>Yes</td>
<td>The ID of the user who created the room.</td>
</tr>
<tr>
<td>SUBJECT</td>
<td>VARCHAR (255)</td>
<td>Yes</td>
<td>The current subject for the room.</td>
</tr>
<tr>
<td>TYPE</td>
<td>VARCHAR (32)</td>
<td>Yes</td>
<td>The constraint check_type. This value must be either &quot;ad-hoc&quot; or &quot;persistent&quot;.</td>
</tr>
</tbody>
</table>
TC_USERS Table

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

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype</th>
<th>Not Null</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOM_JID</td>
<td>VARCHAR (3071)</td>
<td>Yes</td>
<td>The ID of the room.</td>
</tr>
<tr>
<td>REAL_JID</td>
<td>VARCHAR (3071)</td>
<td>Yes</td>
<td>The ID of a user in the room. This value is the actual ID of the user, rather than an alternate name.</td>
</tr>
<tr>
<td>ROLE</td>
<td>VARCHAR (32)</td>
<td>Yes</td>
<td>The role of the user in the room. This value is constrained to one of the following: &quot;none&quot;, &quot;hidden&quot;, &quot;visitor&quot;, &quot;participant&quot;, or &quot;moderator&quot;.</td>
</tr>
<tr>
<td>AFFILIATION</td>
<td>VARCHAR (32)</td>
<td>Yes</td>
<td>The affiliation of the user in the room. This value is constrained to one of the following: &quot;none&quot;, &quot;outcast&quot;, &quot;member&quot;, &quot;admin&quot;, or &quot;owner&quot;.</td>
</tr>
<tr>
<td>NICK_JID</td>
<td>VARCHAR (3071)</td>
<td>Yes</td>
<td>The ID of the room, plus the alternate name for the user. The format is room@tc-server/nick.</td>
</tr>
<tr>
<td>REASON</td>
<td>VARCHAR (255)</td>
<td>Yes</td>
<td>The reason entered when the user's affiliation was last changed.</td>
</tr>
<tr>
<td>INITIATOR_JID</td>
<td>VARCHAR (3071)</td>
<td>Yes</td>
<td>The ID of the room in which the configuration change occurred.</td>
</tr>
</tbody>
</table>
**TC_MESSAGES Table**

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

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype</th>
<th>Not Null</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSG_ID</td>
<td>BIGINT</td>
<td>Yes</td>
<td>The ID of the message. The MSG_ID is a unique identifier for each message per chat room; it is not globally unique.</td>
</tr>
<tr>
<td>ROOM_JID</td>
<td>VARCHAR (3071)</td>
<td>Yes</td>
<td>The ID of the room to which the message was sent.</td>
</tr>
<tr>
<td>STAMP</td>
<td>TIMESTAMP</td>
<td>Yes</td>
<td>The date and time the message was sent.</td>
</tr>
<tr>
<td>MSG</td>
<td>TEXT</td>
<td>Yes</td>
<td>The entire message.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype</th>
<th>Not Null</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REAL_JID</td>
<td>VARCHAR (3071)</td>
<td>Yes</td>
<td>The ID of the user who is entering or leaving the room.</td>
</tr>
<tr>
<td>NICK_JID</td>
<td>VARCHAR (3071)</td>
<td>Yes</td>
<td>The ID of the room, plus the alternate name for the user. The format is room@tc-server/nick.</td>
</tr>
<tr>
<td>DIRECTION</td>
<td>VARCHAR (1)</td>
<td>Yes</td>
<td>Indicates whether the user entered (E) or left (L) the room. Constrained to the values “E” and “L”.</td>
</tr>
<tr>
<td>STAMP</td>
<td>TIMESTAMP</td>
<td>Yes</td>
<td>The date and time at which the user entered or left the room.</td>
</tr>
</tbody>
</table>

**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 the IM and Presence Service. Choose the option Archive all room messages on the Cisco Unified CM IM and Presence Administration user interface under **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.
**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.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype</th>
<th>Not Null</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSG_ID</td>
<td>BIGINT</td>
<td>Yes</td>
<td>A unique identifier for the message.</td>
</tr>
<tr>
<td>TO_JID</td>
<td>VARCHAR (3071)</td>
<td>Yes</td>
<td>The ID of the room that received the message.</td>
</tr>
<tr>
<td>FROM_JID</td>
<td>VARCHAR (3071)</td>
<td>Yes</td>
<td>The ID of the user who sent the message.</td>
</tr>
<tr>
<td>NICK_JID</td>
<td>VARCHAR (3071)</td>
<td>Yes</td>
<td>The ID of the room, plus the alternate name of the sender; for example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="mailto:room@conference.exmpl.com">room@conference.exmpl.com</a>/nick</td>
</tr>
<tr>
<td>SENT_DATE</td>
<td>TIMESTAMP</td>
<td>Yes</td>
<td>The date on which the message was sent.</td>
</tr>
<tr>
<td>MSG_TYPE</td>
<td>VARCHAR (1)</td>
<td>Yes</td>
<td>The first character of the type attribute of the message. The possible</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>values are &quot;c&quot; (chat), &quot;n&quot; (normal), &quot;g&quot; (groupchat), &quot;h&quot; (headline), and</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;e&quot; (error).</td>
</tr>
<tr>
<td>BODY_LEN</td>
<td>INT</td>
<td>Yes</td>
<td>The length in characters of the message body.</td>
</tr>
<tr>
<td>MESSAGE_LEN</td>
<td>INT</td>
<td>Yes</td>
<td>The length in characters of the message, including the subject and body.</td>
</tr>
<tr>
<td>BODY_STRING</td>
<td>VARCHAR (4000)</td>
<td>Yes</td>
<td>The message body.</td>
</tr>
<tr>
<td>MESSAGE_STRING</td>
<td>VARCHAR (4000)</td>
<td>Yes</td>
<td>The entire raw packet.</td>
</tr>
<tr>
<td>BODY_TEXT</td>
<td>TEXT</td>
<td>Yes</td>
<td>If the message body exceeds 4000 characters, it is stored in this field</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>rather than the BODY_STRING field.</td>
</tr>
<tr>
<td>MESSAGE_TEXT</td>
<td>TEXT</td>
<td>Yes</td>
<td>If the entire raw packet exceeds 4000 characters, it is stored in this</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>column rather than in the MESSAGE_STRING column.</td>
</tr>
<tr>
<td>SUBJECT</td>
<td>VARCHAR (255)</td>
<td>Yes</td>
<td>The subject of a message.</td>
</tr>
<tr>
<td>Column Name</td>
<td>Datatype</td>
<td>Not Null</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SUBJECT</td>
<td>VARCHAR (128)</td>
<td>Yes</td>
<td>The subject line of the message that is being archived.</td>
</tr>
<tr>
<td>THREAD_ID</td>
<td>VARCHAR (128)</td>
<td>Yes</td>
<td>The thread ID of the message that is being archived.</td>
</tr>
<tr>
<td>MSG_TYPE</td>
<td>VARCHAR (1)</td>
<td>Yes</td>
<td>The first character of the message's type attribute. The possible values are &quot;c&quot; (chat), &quot;n&quot; (normal), &quot;g&quot; (groupchat), &quot;h&quot; (headline), and &quot;e&quot; (error).</td>
</tr>
<tr>
<td>DIRECTION</td>
<td>VARCHAR (1)</td>
<td>Yes</td>
<td>Indicates whether the message is &quot;O&quot; (Outgoing) or &quot;I&quot; (Incoming). If the message is sent between users on the same node, it is logged twice: once as Outgoing and once as Incoming.</td>
</tr>
<tr>
<td>BODY_LEN</td>
<td>INT</td>
<td>Yes</td>
<td>The number of characters in the message body.</td>
</tr>
<tr>
<td>MESSAGE_LEN</td>
<td>INT</td>
<td>Yes</td>
<td>The number of characters in the message, including the subject and the body.</td>
</tr>
<tr>
<td>BODY_STRING</td>
<td>VARCHAR (4000)</td>
<td>Yes</td>
<td>The message body.</td>
</tr>
<tr>
<td>MESSAGE_STRING</td>
<td>VARCHAR (4000)</td>
<td>Yes</td>
<td>The entire raw packet.</td>
</tr>
<tr>
<td>BODY_TEXT</td>
<td>TEXT</td>
<td>Yes</td>
<td>If the message body exceeds 4000 characters, it is stored in this field rather than the BODY_STRING field.</td>
</tr>
<tr>
<td>MESSAGE_TEXT</td>
<td>TEXT</td>
<td>Yes</td>
<td>If the entire raw packet exceeds 4000 characters, it is stored in this field rather than in the MESSAGE_STRING field.</td>
</tr>
<tr>
<td>HISTORY_FLAG</td>
<td>VARCHAR (1)</td>
<td>Yes</td>
<td>Used when room history messages are sent to new participants (upon entering an existing room). This allows you to distinguish between messages received while actively participating in a room and those received as part of a history push. The latter message type is flagged with HISTORY_FLAG=&quot;H&quot; in the database. Otherwise, this column is set to &quot;N.&quot;</td>
</tr>
</tbody>
</table>

**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:

```sql
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%';