Oracle Utilities Mobile Workforce Management

Database Administrator's Guide

Release 2.3.0.2

E91542-01

February 2018
(Updated May 2020)

ORACLE®

Oracle Utilities Mobile Workforce Management Database Administrator's Guide, Release 2.3.0.2

Contents

Chapter 4

Chapter 5

Appendix A

Appendix B

Appendix C

Appendix D

Appendix E

Preface

This guide provides instructions for installing and maintaining the database for Oracle Utilities Mobile Workforce Management.

This preface contains these topics:

- Audience

- Related Documents

- Updates to this Documentation

- Conventions

Audience

Oracle Utilities Mobile Workforce Management Database Administrator's Guide is intended for database administrators who will be installing and maintaining the database for Oracle Utilities Mobile Workforce Management.

Related Documents

For more information, see these Oracle documents:

Installation, Configuration, and Release Notes

- *Oracle Utilities Mobile Workforce Management Release Notes*

- *Oracle Utilities Mobile Workforce Management Quick Install Guide*

- *Oracle Utilities Mobile Workforce Management Server Application Installation Guide*

- *Oracle Utilities Mobile Workforce Management DBA Guide*

- *Oracle Utilities Mobile Workforce Management Hybrid Mobile Application Installation and Deployment Guide*

User Guides

- *Oracle Utilities Mobile Workforce Management Administrative User Guide*

- *Oracle Utilities Mobile Workforce Management Business User Guide*

- *Oracle Utilities Mobile Workforce Management Mobile Application User's Guide (Java-based)*

- *Oracle Utilities Mobile Workforce Management Hybrid Mobile Application User's Guide*

- *Oracle Utilities Mobile Workforce Management Hybrid Mobile Contractor Application User's Guide*

Map Editor Installation and User Guides
- *Oracle Utilities Mobile Workforce Management Map Editor User's Guide*

- *Oracle Utilities Mobile Workforce Management Map Editor Installation Guide*

Supplemental Documents
- *Oracle Utilities Mobile Workforce Management Server Administration Guide*

- *Oracle Utilities Mobile Workforce Management Security Guide*

Updates to this Documentation

This documentation is provided with the version of the product indicated. Additional and updated information about the operations and configuration of the product is available from the Knowledge Base section of My Oracle Support (http://support.oracle.com). Please refer to My Oracle Support for more information.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Chapter 1

Database Overview

This chapter provides an overview of the database installation for Oracle Utilities Mobile Workforce Management.

This section includes the following information:

- Supported Database Platforms
- Database Maintenance Rules

Supported Database Platforms

This section defines the platforms on which Oracle Utilities Mobile Workforce Management is verified to operate.

Supported Platforms Summary Table

Oracle Utilities Mobile Workforce Management is certified on the following platforms:

Platform	Database Version
AIX 7.1 TL01/ AIX 7.2 TL0 (POWER 64-bit)	Oracle Database Server 12.1.0.1+ (64-bit)/ 12.2.0.1+(64-bit)
Oracle Enterprise Linux 6.x or 7.x (64-bit) x86_64 (64-bit)	Oracle Database Server 12.1.0.1+ (64-bit)/ 12.2.0.1+(64-bit)
Red Hat Enterprise Linux 6.x or 7.x (64-bit) x86_64 (64-bit	Oracle Database Server 12.1.0.1+ (64-bit)/ 12.2.0.1+(64-bit)
Oracle Solaris 11 (SPARC 64-bit)	Oracle Database Server 12.1.0.1+ (64-bit)/ 12.2.0.1+(64-bit)
Windows Server 2008 R2 Windows Server 2012 R2 (x86_64 64-bit)	Oracle Database Server 12.1.0.1+ (64-bit)/ 12.2.0.1+(64-bit)

> **Note:** Oracle Utilities Mobile Workforce Management is tested on both Oracle Database Enterprise Edition and Standard Edition. Some features, such as Advanced Compression and Partitioning, require the Enterprise Edition.
>
> While creating the 12.2.0.1 database make sure the Compatible parameter is set to 12.1.0.2.

Refer to My Oracle Support for additional details.

Support for Software Patches and Upgrades

Due to the ongoing nature of software improvement, vendors will periodically issue patches and service packs for the operating systems, application servers and database servers on top of specific versions that Oracle products have already been tested against.

If it is necessary to apply an upgrade, please do so in a test environment that is running on the same platform as your production environment prior to updating the production environment itself. The exception to this is Hibernate software 4.1.0 which should not be upgraded.

Always contact Oracle Support prior to applying vendor updates that do not guarantee backward compatibility.

Database Maintenance Rules

The database supplied with the product consists of the following elements:

- A set of users to administrate, execute and read the database schema provided.

- A set of database roles to implement security for each of the users provided.

- A tablespace and a schema containing the base database objects used by the product.

The installation instructions are outlined in the installation section of this document.

Permitted Database Changes

During and after installation of the product the following changes may be performed by the database administrator personnel on site:

- Users supplied by product may be changed according to the site standards.

- Database objects may be added to the schema according to database naming standards outlined later in this document.

- Database views and indexes may be created against base database objects. Please make sure to prefix new items with "CM" (for customer modification).

- Database storage attributes for base indexes and base tables may be changed according to site standards and hardware used.

- Tablespace names, attributes and locations may be changed according to site standards.

- Database topology (base table/index to tablespace, tablespace to data file, data file to location) may be altered according to tuning and/or site standards.

- Database triggers may be created against base database objects unless they attempt to contravene base data integrity rules.

- Database initialization and parameter settings may be altered according to site standards unless otherwise advised by Oracle Support or outlined.

Non-Permitted Database Changes

In order to maintain operability and upgradeability of the product, during and after the installation of the product, the following changes may *not* be performed by the database administration personnel on site.

Base objects must not be removed or altered in the following ways:

- Columns in base tables must not be altered, removed or added in anyway.

- Columns in Indexes must not be altered or removed.

- Tables must not be renamed or removed.

- Base views must not be renamed or removed.

- Base Triggers and Sequences must not be renamed or removed.

- Base indexes must not be altered or removed.

Chapter 2

Database Installation Overview

This section provides the steps required to install or upgrade the Oracle Utilities Mobile Workforce Management database, including:

- Installation Overview

- Initial Install

- Upgrade Install

- Demo Install

- Accelerator Install

Installation Overview

Refer to Supported Database Platforms for information about the supported platforms on which Oracle Utilities Mobile Workforce Management is verified to operate.

The following types of installation are available:

- Initial Install - a database with no demo data.

- Upgrade Install - a database upgrade to version 2.3.0.2

- Demo Install - a database populated with demo data.

- Accelerator Install - a database populated with control data.

Pre-Requisites

The database installation requires:

- Java Development Kit Version 8.0

- Oracle Database 12c Release 1 Client (12.1.0.1+) - 32-bit

  This must be installed on the Windows desktop where the install package is staged and run.

Creating the Database

For an initial install or demo install you will create an empty database on the Unix or Windows database server on which you operate the production instance of Oracle Utilities Mobile Workforce Management.

1. Create the database using the Database Configuration Assistant (DBCA). Refer to https://docs.oracle.com/database/121/ADMIN/create.htm#ADMIN12479 for more information related to 12C Release 1. Make sure to set character set for database as AL32UTF8.

   Note: While prior versions of the product have included the cdxdba programs (cdxdba.plx for UNIX or CDXDBA.exe for Windows), this is no longer supported going forward, and the Database Configuration Assistant should be used instead.

   Please refer to Oracle database documentation for the installation steps of the respective version.

2. Enable Mandatory Software Options (as per the licenses acquired):

   - Oracle Spatial OR Oracle Locator

   - Oracle Text

3. Run following SQL to make sure it is successful

   ```
   SELECT COMP_NAME,STATUS FROM DBA_REGISTRY WHERE COMP_NAME IN
   ('Spatial','Oracle Text');
   ```

4. Create default tablespace CISTS_01 and required users and roles as follows.

   ```
   CREATE TABLESPACE CISTS_01 LOGGING DATAFILE '/<db_file_location>/
   cists01.dbf' SIZE 1024M REUSE AUTOEXTEND ON NEXT 8192K MAXSIZE
   UNLIMITED EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1M;
   ```

5. Create required roles as follows:

   ```
   CREATE ROLE CIS_USER;
   CREATE ROLE CIS_READ;
   ```

6. Create users as follows:

```
CREATE USER CISADM IDENTIFIED BY CISADM DEFAULT TABLESPACE CISTS_01
TEMPORARY TABLESPACE TEMP PROFILE DEFAULT;
GRANT UNLIMITED TABLESPACE TO CISADM WITH ADMIN OPTION;
GRANT SELECT ANY TABLE TO CISADM;
GRANT CREATE DATABASE LINK TO CISADM;
GRANT CONNECT TO CISADM;
GRANT RESOURCE TO CISADM;
GRANT DBA TO CISADM WITH ADMIN OPTION;
GRANT CREATE ANY SYNONYM TO CISADM;
GRANT SELECT ANY DICTIONARY TO CISADM;

CREATE USER CISUSER PROFILE DEFAULT IDENTIFIED BY CISUSER DEFAULT
TABLESPACE CISTS_01 TEMPORARY TABLESPACE TEMP;
GRANT SELECT ANY TABLE TO CISUSER;
GRANT CIS_USER TO CISUSER;
GRANT CIS_READ TO CISUSER;
GRANT CONNECT TO CISUSER;

CREATE USER CISOPR PROFILE DEFAULT IDENTIFIED BY OPRPLUS DEFAULT
TABLESPACE CISTS_01 TEMPORARY TABLESPACE TEMP;
GRANT CONNECT,RESOURCE,EXP_FULL_DATABASE TO CISOPR;

CREATE USER CISREAD IDENTIFIED BY CISREAD DEFAULT TABLESPACE
CISTS_01 TEMPORARY TABLESPACE TEMP;
GRANT SELECT ANY TABLE TO CISREAD;
GRANT CIS_READ TO CISREAD;
GRANT CONNECT TO CISREAD;
```

7. Review the Storage.xml file under the FW43040\Install-Upgrade folder prior to an initial install or upgrade install. This file allocates all base tables and indexes to the default tablespace CISTS_01 and the required users and roles. Information in this file is used by ORADBI while installing the Oracle Utilities Mobile Workforce Management database objects. Refer to Updating Storage.xml for more details on updating this file.

> **Note:** You will need to review the Storage.xml file, prior to an initial install, to update the default values to custom values (for e.g. TableSpace Name). OraDBI can be executed by a non-schema owner in order to upgrade the database. The Initial Install still needs to be done by the schema owner.

If you decide to allocate some tables or indexes outside of the default tablespace, change the tablespace name from the default value to a custom value in the Storage.xml file.

For instance, if you decide to allocate table CI_ACCT in a tablespace MyTablespace, change Storage.xml as shown:

```
<CI_ACCT>
<TABLESPACE>MyTablespace</TABLESPACE>
</CI_ACCT>
```

For optimum storage allocation, database administrators should create multiple tablespaces with extents sized to store different types of tables/indexes. They can then edit the storage.xml file before install process, to spread tables and indexes across these tablespaces. Tables and indexes can be created in parallel by editing degree of parallelism. Tablespace, storage options, secure file options, Advanced Compression, and parallel information are used only for new objects. Therefore, for initial installs, information for each object should be reviewed. Be careful while editing this file. Make sure that tablespace names being used exist in the database. Do not change the basic format of this file.

> **Note:** Prior to the installation of the database schema for the product, please ensure that the Database Management System software is installed according to

your site standards and the installation guide provided by the database vendor. Also please make sure that you have necessary licenses to use some of the advanced database features such as Advanced Compression.

Initial Install

This section describes how to install the database components of Oracle Utilities Mobile Workforce Management, including:

- Copying and Decompressing Install Media

- Database Creation

- Installing the ORSMWM Schema

- Postinstallation Tasks

- Installing Service Packs and Patches

Copying and Decompressing Install Media

To copy and decompress the Oracle Utilities Mobile Workforce Management database:

1. Download Oracle Utilities Mobile Workforce Management v2.3.0.2 from the Oracle Software Delivery Cloud.

2. Unzip the MWM-V2.3.0.2.0-database.zip files to a temporary folder. The Database folder contains several files that will be referred to in the installation process.

Database Creation

> **Note:** This step is not required if you are performing a database upgrade from a previous version of the product.
>
> You must have Oracle Database Server 12.1.0.1+ or Oracle Database Server 12.2.0.1+ on your machine in order to create the database.
>
> For 12.2.0.1 database set the compatible parameter to 12.1.0.2 before creating the database.

Creating the Database on UNIX

Create the database using the Database Configuration Assistant (DBCA).

Refer to https://docs.oracle.com/database/121/ADMIN/create.htm#ADMIN12479 for more information about 12c Release 1 database installation. Make sure to set character set for database as AL32UTF8.

> **Note**: Follow the Oracle database documentation for steps to create the database of respective version.

Refer to Creating the Database for steps to create the database.

Creating the Database on Windows

You should be logged in as a user who is a member of the local ORA_DBA group on that server. The ORA_DBA group should have "administrator" privileges assigned to it.

Refer to https://docs.oracle.com/database/121/ADMIN/create.htm#ADMIN12479 for more information about 12c Release 1 database installation. Make sure to set character set for database as AL32UTF8.

> **Note**: Follow the Oracle database documentation for steps to create the database of respective version.

Refer to Creating the Database for steps to create the database.

Installing the MWM Schema

You will install Oracle Utilities Application Framework v4.3.0.4 and Oracle Real-Time Scheduler prior to Oracle Utilities Mobile Workforce Management 2.3.0.2. The files for Oracle Utilities Application Framework installation are located in the FW folder for the specific versions.

Installing the Oracle Utilities Application Framework Database Component using OraDBI.jar

Prepare the following parameters before installation:

- The name of the database server in which the database is configured- DB_SERVER

- The listener port number of the database - PORT

- The target database name in which the product is to be installed - SID

- A database user that will own the application schema (for example, CISADM) - DBUSER

- Password of the database user that will own the application schema - DBPASS

- A database user that has read-write (select/update/insert/delete) privileges to the objects in the application schema (for example, CISUSER). The application will access the database as this user - RWUSER

- A database user with read-only privileges to the objects in the application schema (for example, CISREAD) - RUSER

- A database role that has read-write (select/update/insert/delete) privileges to the objects in the application schema (for example, CIS_USER) - RW_USER_ROLE

- A database role with read-only privileges to the objects in the application schema (for example, CIS_READ) - R_USER_ROLE

- Location for jar files (the Jar files are bundled with the database package) - CLASSPATH

- Java Home (for example, C:\Java\jdk1.8.0) - JAVA_HOME

You can execute OraDBI.jar using either of the following methods:

- Using Interactive Mode

- Using Command Line Mode

Using Interactive Mode

The following procedure lists the steps to install the schema for Oracle Utilities Application Framework v4.3.0.4.0 using OraDBI.

Run the following command with the defined parameters on the command prompt from the FW\FW43040\Install-Upgrade directory.

1. Open a command line prompt in the Windows environment.

2. Set Java Home.

   In the following example, JDK 1.8 is installed in C:\Program Files\Java\jdk1.8.0_101 directory.

   ```
   SET JAVA_HOME=C:\Program Files\Java\jdk1.8.0_101
   ```

3. Set the class path.

    In the following example, the required jarfiles, including OraDBI.jar, are available in the C:\InstallUpgrade\Jarfiles* directory.

    ```
    SET CLASSPATH=C:\InstallUpgrade\Jarfiles\*
    ```

4. Execute the following command:

    ```
    "%JAVA_HOME%"\bin\java -Xmx1500M -cp %CLASSPATH%
    com.oracle.ouaf.oem.install.OraDBI

        (or)

    "C:\Program Files\Java\jdk1.8.0_101"\bin\java -Xmx1500M -cp
    C:\InstallUpgrade\Jarfiles\* com.oracle.ouaf.oem.install.OraDBI
    ```

    The utility prompts you to enter values for the following parameters as per your environment:

    *   Name of the database server: <DB SERVER>

    *   Port no: <PORT>

    *   Name of the target database: <SID>

    *   Name of the owner of the database schema: <DBUSER>

    *   Password of the user name: <DBPASS>

    *   Location of Java Home: (e.g. C:\Java\jdk1.8.0): <Java Home>

    *   Oracle user with read-write privileges to the Database Schema: <CISUSER>

    *   Oracle user with read-only privileges to the Database Schema: <CISREAD>

    *   Oracle database role with read-write privileges to the Database Schema: <CIS_USER>

    *   Oracle database role with read-only privileges to the Database Schema: <CIS_READ>

    *   Enter the name of the target schema where you want to install or upgrade: <CISADM>

    *   Enter the password for the target schema: <CISADM password>

This process generates log files in the directory Install-Upgrade\logs. Make sure to check log files for any errors.

> **Note**: For OraDBI jar, you may receive the following message in the display output or logs. These errors can be safely ignored and the process should proceed to completion.

```
- 2016-05-23 16:31:38,315 [main] ERROR
(common.cryptography.KeyStoreWrapperFactory) The keystore file
'<filename>' does not exist.…
This file is either provided by the property
com.oracle.ouaf.system.keystore.file or expected to exist at the
default file location null Attempting to use the legacy
cryptography.
 - 2016-05-23 16:31:38,566 [main] INFO  (oem.install.OraDBI)
```

You are now ready to install the Oracle Real-Time Scheduler database component. Please proceed with the steps in the section Installing the Oracle Real-Time Scheduler Database Component.

Using Command Line Mode

Run the following command with the defined parameters on the command prompt from .. FW\FW43040\Install-Upgrade directory.

```
"C:\Program Files\Java\jdk1.8.0_65\bin\java" -Xmx1500M -cp
C:\InstallUpgrade\Jarfiles\* com.oracle.ouaf.oem.install.OraDBI -d
jdbc:oracle:thin:@DB_Server:1521/SID,
DBUSER,DBPASS,RWUSER,RUSER,RW_USER_ROLE,R_USER_ROLE,DBUSER -l 1,2 -
j "C:\Program Files\Java\jdk1.8.0_65" C:\InstallUpgrade\Jarfiles -q
true
```

This process generates log files in the directory Install-Upgrade\logs. Make sure to check log files for any errors.

> **Note**: For OraDBI jar, you may receive the following message in the display output or logs. These errors can be safely ignored and the process should proceed to completion.

```
-  2016-05-23 16:31:38,315 [main] ERROR
(common.cryptography.KeyStoreWrapperFactory) The keystore file
'<filename>' does not exist.…
This file is either provided by the property
com.oracle.ouaf.system.keystore.file or expected to exist at the
default file location null Attempting to use the legacy
cryptography.
 -  2016-05-23 16:31:38,566 [main] INFO  (oem.install.OraDBI)
```

Installing Prerequisite Database Single Fixes

Before installing Oracle Utilities Mobile Workforce Management, you must install Oracle Utilities Framework Prerequisite DB Hot Fixes.

> **Note**: While prior versions of the product have included the cdxpatch.exe programs for applying DB Hot Fixes, this is no longer supported going forward, and the ouafDatabasePatch.cmd or ouafDatabasePatch.sh should be used instead.

To install the Framework Prerequisite DB Hot Fixes, follow these steps:

Applying Hot Fixes from a Windows machine:

> **Note**: You must have Java 8 JDK installed on the machine to use the command. Be sure to install the JDK that is supported for your platform.

1. Copy the FW\FW43040-Rollup\db_patch_standalone.jar to a directory on Windows, under c:\dbpatch_tools and extract the db_patch_standalone.jar or use the following command:

    ```
    cd c:\dbpatch_tools
    jar xvf db_patch_standalone.jar
    ```

2. SET TOOLSBIN=c:\dbpatch_tools\bin

3. Apply prerequisite Framework DB single fixes by running the ouafDatabasePatch.cmd utility from the ..\FW\FW43040-Rollup\Database directory. The utility will prompt you for the value of the following parameters:

    • Enter the target database type (O/M/D) [O]:

    • Enter the username that owns the schema: <CISADM>

- Enter the password for the cisadm user: <CISADM Password>

- Enter the name of the Oracle Database Connection String: <DB_Server:DBPORT:ORACLE_SID> > or <//DB_Server:DBPORT/ SERVICE_NAME>

Applying Hotfixes from a Unix Standalone server:

> **Note**: You must have Java 8 JDK installed on the machine to use the command. Be sure to install the JDK that is supported for your platform.

1. Copy the .\FW\FW43040-Rollup\db_patch_standalone.jar to a directory on Unix server, under /tmp/dbpatch_tools and extract the db_patch_standalone.jar using below command:

```
cd /tmp/dbpatch_tools
jar xvf db_patch_standalone.jar
```

2. Export TOOLSBIN=/tmp/dbpatch_tools/bin

3. Apply prerequisite Framework DB single fixes by running the ouafDatabasePatch.sh utility from the ..\FW\FW43040-Rollup\Database directory. The utility prompts you for the value of the following parameters:

- Enter the target database type (O/M/D) [O]

- Enter the username that owns the schema: <CISADM>

- Enter the password for the cisadm user: <CISADM Password>

- Enter the name of the Oracle Database Connection String: <DB_Server:DBPORT:ORACLE_SID> or <//DB_Server:DBPORT/ SERVICE_NAME>

Installing the Oracle Real-Time Scheduler Database Component

Follow the procedure below to install the database component of Oracle Real-Time Scheduler.

Prepare the following parameters before installation:

- The name of the database server in which the database is configured- DB_SERVER

- The listener port number of the database - PORT

- The target database name in which the product is to be installed - SID

- A database user that will own the application schema (for example, CISADM) - DBUSER

- Password of the database user that will own the application schema - DBPASS

- A database user that has read-write (select/update/insert/delete) privileges to the objects in the application schema (for example, CISUSER). The application will access the database as this user - RWUSER

- A database user with read-only privileges to the objects in the application schema. (for example, CISREAD) - RUSER

- A database role that has read-write (select/update/insert/delete) privileges to the objects in the application schema. (for example, CIS_USER) - RW_USER_ROLE

- A database role with read-only privileges to the objects in the application schema. (for example, CIS_READ)- R_USER_ROLE

- Location for jar files. (The Jar files are bundled with the database package) - CLASSPATH

- Java Home (for example, C:\Java\jdk1.8.0) - JAVA_HOME

You can execute OraDBI.jar using either of the following methods:

-

-

Using Interactive Mode

The following procedure lists the steps to install the schema for Oracle Real-Time Scheduler v2.3.0.2 using OraDBI.

Run the following command with the defined parameters on the command prompt from .. ORS\Install-Upgrade directory:

1. Open a command line prompt in the Windows environment.

2. Set Java Home.

   In the following example, JDK 1.8 is installed in C:\Program Files\Java\jdk1.8.0_101 directory.

   ```
   SET JAVA_HOME=C:\Program Files\Java\jdk1.8.0_101
   ```

3. Set the class path.

   In the following example, the required jarfiles, including OraDBI.jar, are available in the C:\InstallUpgrade\Jarfiles* directory.

   ```
   SET CLASSPATH=C:\InstallUpgrade\Jarfiles\*
   ```

4. Execute the following command:

   ```
   "%JAVA_HOME%"\bin\java -Xmx1500M -cp %CLASSPATH%
   com.oracle.ouaf.oem.install.OraDBI

       (or)

   "C:\Program Files\Java\jdk1.8.0_101"\bin\java -Xmx1500M -cp
   C:\InstallUpgrade\Jarfiles\* com.oracle.ouaf.oem.install.OraDBI
   ```
   The utility prompts you to enter values for the following parameters as per your environment:

   - Name of the database server: <DB SERVER>

   - Port no: <PORT>

   - Name of the target database: <SID>

   - Name of the owner of the database schema: <DBUSER>

   - Password of the user name: <DBPASS>

   - Location of Java Home: (e.g. C:\Java\jdk1.8.0): <Java Home>

   - Oracle user with read-write privileges to the Database Schema: <CISUSER>

   - Oracle user with read-only privileges to the Database Schema: <CISREAD>

   - Oracle database role with read-write privileges to the Database Schema: <CIS_USER>

   - Oracle database role with read-only privileges to the Database Schema: <CIS_READ>

   - Enter the name of the target schema where you want to install or upgrade: <CISADM>

   - Enter the password for the target schema: <CISADM password>

This process generates log files in the directory Install-Upgrade\logs. Make sure to check log files for any errors.

Note: For OraDBI jar, you may receive the following message in the display output or logs. These errors can be safely ignored and the process should proceed to completion.

```
-  2016-05-23 16:31:38,315 [main] ERROR
(common.cryptography.KeyStoreWrapperFactory) The keystore file
'<filename>' does not exist.…
This file is either provided by the property
com.oracle.ouaf.system.keystore.file or expected to exist at the
default file location null Attempting to use the legacy
cryptography.
 -  2016-05-23 16:31:38,566 [main] INFO  (oem.install.OraDBI)
```

Oracle Real-Time Scheduler database component is now installed.

Using Command Line Mode

Run the following command with the defined parameters on the command prompt from .. ORS\Install-Upgrade directory

```
"C:\Program Files\Java\jdk1.8.0_65\bin\java" -Xmx1500M -cp
C:\InstallUpgrade\Jarfiles\* com.oracle.ouaf.oem.install.OraDBI -d
jdbc:oracle:thin:@DB_Server:1521/SID,
DBUSER,DBPASS,RWUSER,RUSER,RW_USER_ROLE,R_USER_ROLE,DBUSER -l 1,2 -
j "C:\Program Files\Java\jdk1.8.0_65" C:\InstallUpgrade\Jarfiles -q
true
```

This process generates log files in the directory Install-Upgrade\logs. Make sure to check log files for any errors.

Note: For OraDBI jar, you may receive the following message in the display output or logs. These errors can be safely ignored and the process should proceed to completion.

```
-  2016-05-23 16:31:38,315 [main] ERROR
(common.cryptography.KeyStoreWrapperFactory) The keystore file
'<filename>' does not exist.…
This file is either provided by the property
com.oracle.ouaf.system.keystore.file or expected to exist at the
default file location null Attempting to use the legacy
cryptography.
 -  2016-05-23 16:31:38,566 [main] INFO  (oem.install.OraDBI)
```

Oracle Real-Time Scheduler database component is now installed.

Installing the CISADM Schema for Oracle Utilities Mobile Workforce Management

The files for the Initial Install of Oracle Utilities Mobile Workforce Management 2.3.0.2 are located in the Install-Upgrade folder under MWM.

Follow the procedure below to install the database component of Oracle Utilities Mobile Workforce Management.

Prepare the following parameters before installation:

- The name of the database server in which the database is configured- DB_SERVER

- The listener port number of the database - PORT

- The target database name in which the product is to be installed - SID

- A database user that will own the application schema (for example, CISADM) - DBUSER

- Password of the database user that will own the application schema - DBPASS

- A database user that has read-write (select/update/insert/delete) privileges to the objects in the application schema (for example, CISUSER). The application will access the database as this user - RWUSER

- A database user with read-only privileges to the objects in the application schema. (for example, CISREAD) - RUSER

- A database role that has read-write (select/update/insert/delete) privileges to the objects in the application schema. (for example, CIS_USER) - RW_USER_ROLE

- A database role with read-only privileges to the objects in the application schema. (for example, CIS_READ)- R_USER_ROLE

- Location for jar files. (The Jar files are bundled with the database package) - CLASSPATH

- Java Home (for example, C:\Java\jdk1.8.0) - JAVA_HOME

You can execute OraDBI.jar using either of the following methods:

- Using Interactive Mode

- Using Command Line Mode

Using Interactive Mode

The following procedure lists the steps to install the schema for Oracle Utilities Mobile Workforce Management v2.3.0.2 using OraDBI.

Run the following command with the defined parameters on the command prompt from .. MWM\Install-Upgrade directory:

1. Open a command line prompt in the Windows environment.

2. Set Java Home.

   In the following example, JDK 1.8 is installed in C:\Program Files\Java\jdk1.8.0_101 directory.

   ```
   SET JAVA_HOME=C:\Program Files\Java\jdk1.8.0_101
   ```

3. Set the class path.

   In the following example, the required jarfiles, including OraDBI.jar, are available in the C:\InstallUpgrade\Jarfiles* directory.

   ```
   SET CLASSPATH=C:\InstallUpgrade\Jarfiles\*
   ```

4. Execute the following command:

   ```
   "%JAVA_HOME%"\bin\java -Xmx1500M -cp %CLASS_PATH%
   com.oracle.ouaf.oem.install.OraDBI

      (or)

   "C:\Program Files\Java\jdk1.8.0_101"\bin\java -Xmx1500M -cp
   C:\InstallUpgrade\Jarfiles\* com.oracle.ouaf.oem.install.OraDBI
   ```
   The utility prompts you to enter values for the following parameters as per your environment:

   - Name of the database server: <DB SERVER>

   - Port no: <PORT>

   - Name of the target database: <SID>

•Name of the owner of the database schema: <DBUSER>

•Password of the user name: <DBPASS>

•Location of Java Home: (e.g. C:\Java\jdk1.8.0): <Java Home>

•Oracle user with read-write privileges to the Database Schema: <CISUSER>

•Oracle user with read-only privileges to the Database Schema: <CISREAD>

•Oracle database role with read-write privileges to the Database Schema: <CIS_USER>

•Oracle database role with read-only privileges to the Database Schema: <CIS_READ>

•Enter the name of the target schema where you want to install or upgrade: <CISADM>

•Enter the password for the target schema: <CISADM password>

This process generates log files in the directory Install-Upgrade\logs. Make sure to check log files for any errors.

> **Note**: For OraDBI jar, you may receive the following message in the display output or logs. These errors can be safely ignored and the process should proceed to completion.

```
-  2016-05-23 16:31:38,315 [main] ERROR
(common.cryptography.KeyStoreWrapperFactory) The keystore file
'<filename>' does not exist.…
This file is either provided by the property
com.oracle.ouaf.system.keystore.file or expected to exist at the
default file location null Attempting to use the legacy
cryptography.
 -  2016-05-23 16:31:38,566 [main] INFO  (oem.install.OraDBI)
```

Oracle Utilities Mobile Workforce Management database component is now installed.

Using Command Line Mode

Run the following command with the defined parameters on the command prompt from .. MWM\Install-Upgrade directory.

```
"C:\Program Files\Java\jdk1.8.0_65\bin\java" -Xmx1500M -cp
C:\InstallUpgrade\Jarfiles\* com.oracle.ouaf.oem.install.OraDBI -d
jdbc:oracle:thin:@DB_Server:1521/SID,
DBUSER,DBPASS,RWUSER,RUSER,RW_USER_ROLE,R_USER_ROLE,DBUSER -l 1,2 -
j "C:\Program Files\Java\jdk1.8.0_65" C:\InstallUpgrade\Jarfiles -q
true
```

This process generates log files in the directory Install-Upgrade\logs. Make sure to check log files for any errors.

> **Note**: For OraDBI jar, you may receive the following message in the display output or logs. These errors can be safely ignored and the process should proceed to completion.

```
-  2016-05-23 16:31:38,315 [main] ERROR
(common.cryptography.KeyStoreWrapperFactory) The keystore file
'<filename>' does not exist.…
This file is either provided by the property
com.oracle.ouaf.system.keystore.file or expected to exist at the
default file location null Attempting to use the legacy
cryptography.
 -  2016-05-23 16:31:38,566 [main] INFO  (oem.install.OraDBI)
```

Oracle Utilities Mobile Workforce Management database component is now installed.

ORADBI Performs the Following Tasks

- Interacts with the user to collect information about the name of Oracle account that will own the application schema (for example, CISADM), password of this account, and the name of the Oracle account that the application user will use (for example, CISUSER), and the name of the Oracle account that will be assigned read-only privileges to the application schema (for example, CISREAD).

- Verifies whether tablespace names already exist in the Storage.xml file (if not, the process will abort).

- Installs the schema, installs the system data, and configures security.

- Maintains upgrade log tables in the database.

- Updates release ID when the upgrade is completed successfully.

- If an error occurs while executing a SQL script or another utility, it logs and displays the error message and allows you to re-execute the current step. Log files OraDBI###.log are created in the same folder as OraDBI and contains all the SQL commands executed against the database along with the results. The log files are incremental so that the results are never overwritten. If warning messages are generated during the upgrade, OraDBI prompts the user at the end of the process. Users should check the log files to verify the warning messages.

- Warning messages are only alerts and do not necessary mean a problem exists.

- Stores the Schema owner and password in the feature configuration table. The password is stored in encrypted format.

Postinstallation Tasks

Importing Styles and Maps

Make sure that these steps are preformed from a Windows based machine.

To import styles and maps in the specified database, follow these steps:

1. Execute ImportSDOThemes.bat under the ORS/Post-Upgrade directory with the following parameters:

   ImportSDOThemes.bat -u CISADM -p <PASSWORD> -c <DATABASE NAME>

2. After importing themes/styles /GEOM Metadata (zz_user_sdo_themes/zz_user_sdo_styles, zz_user_sdo_metadata tables), postImport.sql is invoked from target main schema. This will:

   - Insert new styles/themes/sdo_geom_metadata in USER_SDO_THEMES/ USER_SDO_STYLES/USER_SDO_GEOM_METADATA

   - Drop ZZ_USER_SDO_THEMES, ZZ_USER_SDO_STYLES, ZZ_USER_SDO_GEOM_METADATA tables from target database.

   - Create domain key indexes

Enable USER_LOCK Package

For inbound web services to work the USER_LOCK must be enabled at the database level. This is a one time step. If this is not already enabled please do so using the following steps.

1. Login as SYS user

2. On SQL prompt run:

   @?/rdbms/admin/userlock.sql

3. Grant permission by running the following SQL:

   grant execute on USER_LOCK to public;

Please note that grant can also be made to the database user which the Application connects to only instead of to public. For example, cisuser.

Populating Language Data

At this point, the Oracle Utilities Mobile Workforce Management database is ready for use. Please note that this database contains data in the ENGLISH language only. If you use any other supported language, run the F1-LANG batch program to duplicate the entries for new language records. For more information on running this batch program, refer to the user documentation section "Defining Background Processes."

You can also install the language specific data packages (if available) into the database. Please contact your Oracle support representative to receive information on these packages.

Generating Database Statistics

During an install process, new database objects may be added to the target database. Before starting to use the database, generate the complete statistics for these new objects by using the DBMS_STATS package.

Installing Service Packs and Patches

Periodically, Oracle Utilities releases a service pack of single fixes for products. A service pack is an update to an existing release that includes solutions to known problems and other product enhancements. A service pack is not a replacement for an installation, but a pack consisting of a collection of changes and additions for the installation. The service pack may include changes to be applied to the application server, the database, or both. The service pack includes all files necessary for installing the collection of changes, including installation instructions.

Between services packs, Oracle Utilities releases patches to fix individual bugs. For information on installing patches, see knowledge base article ID 974985.1 on My Oracle Support.

Service packs and patches can be downloaded from My Oracle Support (https://support.oracle.com/).

Upgrade Install

> **Note:** The new MCP version control enhancement requires that a certain upgrade process be followed. For details about this process, see Chapter, "Planning the Installation," in the Oracle Utilities Mobile Workforce Management *Installation Guide.*

This section describes how to upgrade the database components of Oracle Utilities Mobile Workforce Management, including:

- Copying and Decompressing Install Media
- Upgrading the MWMORS Schema
- Installing Service Packs and Patches

Copying and Decompressing Install Media

To copy and decompress the Oracle Utilities Mobile Workforce Management database:

1. Download the Oracle Utilities Mobile Workforce Management v2.3.0.2 from the Oracle Software Delivery Cloud.

2. Unzip the MWM-V2.3.0.2.0-database.zip file to a temporary folder. The Database folder contains several files that will be referred to during the installation process.

Upgrading the MWM Schema

The following upgrade paths are supported. Choose the procedure applicable to your existing environment:

- Upgrading to v2.3.0.2

Upgrading to v2.3.0.2

This release only supports the following direct upgrade paths:

- Upgrade from version 2.3.0 to version 2.3.0.2
- Upgrade from version 2.3.0.1 to version 2.3.0.2

If you are at v2.3.0 or at v2.3.0.1, follow the instructions in this section to upgrade to v2.3.0.2.

> **Note**: You must have Oracle Database Server 12.1.0.1+ or Oracle Database Server 12.2.0.1+ installed on your machine to continue with the procedure below.
>
> For 12.2.0.1 database please set the compatible parameter to 12.1.0.2 before creating the database.

You must install Oracle Utilities Application Framework version v4.3.0.4 prior to upgrading to Oracle Utilities Mobile Workforce Management v2.3.0.2. The files for Oracle Utilities Application Framework installation are located in the FW/FW43040 folder.

Installing the Oracle Utilities Application Framework Database Component using OraDBI.jar

Prepare the following parameters before installation:

- The name of the database server in which the database is configured- DB_SERVER
- The listener port number of the database - PORT
- The target database name in which the product is to be installed - SID

- A database user that will own the application schema (for example, CISADM) - DBUSER

- Password of the database user that will own the application schema - DBPASS

- A database user that has read-write (select/update/insert/delete) privileges to the objects in the application schema (for example, CISUSER). The application will access the database as this user - RWUSER

- A database user with read-only privileges to the objects in the application schema. (for example, CISREAD) - RUSER

- A database role that has read-write (select/update/insert/delete) privileges to the objects in the application schema. (for example, CIS_USER) - RW_USER_ROLE

- A database role with read-only privileges to the objects in the application schema. (for example, CIS_READ)- R_USER_ROLE

- Location for jar files. (The Jar files are bundled with the database package) - CLASSPATH

- Java Home (for example, C:\Java\jdk1.8.0) - JAVA_HOME

You can execute OraDBI.jar using either of the following methods:

- Using Interactive Mode

- Using Command Line Mode

Using Interactive Mode

The following procedure lists the steps to install the schema for Oracle Utilities Application Framework v4.3.0.4.0 using OraDBI.

Run the following command with the defined parameters on the command prompt from the FW\FW43040\Install-Upgrade directory.

1. Open a command line prompt in the Windows environment.

2. Set Java Home.

    In the following example, JDK 1.8 is installed in C:\Program Files\Java\jdk1.8.0_101 directory.

    ```
    SET JAVA_HOME=C:\Program Files\Java\jdk1.8.0_101
    ```

3. Set the class path.

    In the following example, the required jarfiles, including OraDBI.jar, are available in the C:\InstallUpgrade\Jarfiles* directory.

    ```
    SET CLASSPATH=C:\InstallUpgrade\Jarfiles\*
    ```

4. Execute the following command:

    ```
    "%JAVA_HOME%"\bin\java -Xmx1500M -cp %CLASSPATH%
    com.oracle.ouaf.oem.install.OraDBI

        (or)

    "C:\Program Files\Java\jdk1.8.0_101"\bin\java -Xmx1500M -cp
    C:\InstallUpgrade\Jarfiles\* com.oracle.ouaf.oem.install.OraDBI
    ```

The utility prompts you to enter values for the following parameters as per your environment:

- Name of the database server: <DB SERVER>

- Port no: <PORT>

- Name of the target database: <SID>

- Name of the owner of the database schema: <DBUSER>

- Password of the user name: <DBPASS>

- Location of Java Home: (e.g. C:\Java\jdk1.8.0): <Java Home>

- Oracle user with read-write privileges to the Database Schema: <CISUSER>

- Oracle user with read-only privileges to the Database Schema: <CISREAD>

- Oracle database role with read-write privileges to the Database Schema: <CIS_USER>

- Oracle database role with read-only privileges to the Database Schema: <CIS_READ>

- Enter the name of the target schema where you want to install or upgrade: <CISADM>

- Enter the password for the target schema: <CISADM password>

This process generates log files in the directory Install-Upgrade\logs. Make sure to check log files for any errors.

> **Note**: For OraDBI jar, you may receive the following message in the display output or logs. These errors can be safely ignored and the process should proceed to completion.

```
- 2016-05-23 16:31:38,315 [main] ERROR
(common.cryptography.KeyStoreWrapperFactory) The keystore file
'<filename>' does not exist.…
This file is either provided by the property
com.oracle.ouaf.system.keystore.file or expected to exist at the
default file location null Attempting to use the legacy
cryptography.
 - 2016-05-23 16:31:38,566 [main] INFO  (oem.install.OraDBI)
```

You are now ready to install the Oracle Real-Time Scheduler database component. Please proceed with the steps in the section Installing the Oracle Real-Time Scheduler Database Component.

Using Command Line Mode
Run the following command with the defined parameters on the command prompt from .. FW\FW43040\Install-Upgrade directory.

```
"C:\Program Files\Java\jdk1.8.0_65\bin\java" -Xmx1500M -cp
C:\InstallUpgrade\Jarfiles\* com.oracle.ouaf.oem.install.OraDBI -d
jdbc:oracle:thin:@DB_Server:1521/SID,
DBUSER,DBPASS,RWUSER,RUSER,RW_USER_ROLE,R_USER_ROLE,DBUSER -l 1,2 -
j "C:\Program Files\Java\jdk1.8.0_65" C:\InstallUpgrade\Jarfiles -q
true
```

This process generates log files in the directory Install-Upgrade\logs. Make sure to check log files for any errors.

Note: For OraDBI jar, you may receive the following message in the display output or logs. These errors can be safely ignored and the process should proceed to completion.

```
-  2016-05-23 16:31:38,315 [main] ERROR
(common.cryptography.KeyStoreWrapperFactory) The keystore file
'<filename>' does not exist.…
This file is either provided by the property
com.oracle.ouaf.system.keystore.file or expected to exist at the
default file location null Attempting to use the legacy
cryptography.
  -  2016-05-23 16:31:38,566 [main] INFO  (oem.install.OraDBI)
```

Installing Prerequisite Database Single Fixes

Before installing Oracle Utilities Mobile Workforce Management, you must install Oracle Utilities Framework Prerequisite DB Hot Fixes.

> **Note**: While prior versions of the product have included the cdxpatch.exe programs for applying DB Hot Fixes, this is no longer supported going forward, and the ouafDatabasePatch.cmd or ouafDatabasePatch.sh should be used instead.

To install the Framework Prerequisite DB Hot Fixes, follow these steps:

Applying Hot Fixes from a Windows machine:

> **Note**: You must have Java 8 JDK installed on the machine to use the command. Be sure to install the JDK that is supported for your platform.

1. Copy the FW\FW43040-Rollup\db_patch_standalone.jar to a directory on Windows, under c:\dbpatch_tools and extract the db_patch_standalone.jar or use the following command:

   ```
   cd c:\dbpatch_tools
   jar xvf db_patch_standalone.jar
   ```

2. SET TOOLSBIN=c:\dbpatch_tools\bin

3. Apply prerequisite Framework DB single fixes by running the ouafDatabasePatch.cmd utility from the ..\FW\FW43040-Rollup\Database directory. The utility will prompt you for the value of the following parameters:

   • Enter the target database type (O/M/D) [O]:

   • Enter the username that owns the schema: <CISADM>

   • Enter the password for the cisadm user: <CISADM Password>

   • Enter the name of the Oracle Database Connection String: <DB_Server:DBPORT:ORACLE_SID> or <//DB_Server:DBPORT/ SERVICE_NAME>

Applying Hotfixes from a Unix Standalone server:

> **Note**: You must have Java 8 JDK installed on the machine to use the command. Be sure to install the JDK that is supported for your platform.

1. Copy the .\FW\FW43040-Rollup\db_patch_standalone.jar to a directory on Unix server, under /tmp/dbpatch_tools and extract the db_patch_standalone.jar using below command:

   ```
   cd /tmp/dbpatch_tools
   jar xvf db_patch_standalone.jar
   ```

2. Export TOOLSBIN=/tmp/dbpatch_tools/bin

3. Apply prerequisite Framework DB single fixes by running the ouafDatabasePatch.sh utility from the ..\FW\FW43040-Rollup\Database directory. The utility prompts you for the value of the following parameters:

- Enter the target database type (O/M/D) [O]:

- Enter the username that owns the schema: <CISADM>

- Enter the password for the cisadm user: <CISADM Password>

- Enter the name of the Oracle Database Connection String: <DB_Server:DBPORT:ORACLE_SID> or <//DB_Server:DBPORT/ SERVICE_NAME>

Installing the Oracle Real-Time Scheduler Database Component

Follow the procedure below to install the database component of Oracle Real-Time Scheduler.

Prepare the following parameters before installation:

- The name of the database server in which the database is configured- DB_SERVER

- The listener port number of the database - PORT

- The target database name in which the product is to be installed - SID

- A database user that will own the application schema (for example, CISADM) - DBUSER

- Password of the database user that will own the application schema - DBPASS

- A database user that has read-write (select/update/insert/delete) privileges to the objects in the application schema (for example, CISUSER). The application will access the database as this user - RWUSER

- A database user with read-only privileges to the objects in the application schema. (for example, CISREAD) - RUSER

- A database role that has read-write (select/update/insert/delete) privileges to the objects in the application schema. (for example, CIS_USER) - RW_USER_ROLE

- A database role with read-only privileges to the objects in the application schema. (for example, CIS_READ) - R_USER_ROLE

- Location for jar files. (The Jar files are bundled with the database package) - CLASSPATH

- Java Home (for example, C:\Java\jdk1.8.0) - JAVA_HOME

You can execute OraDBI.jar using either of the following methods:

- Using Interactive Mode

- Using Command Line Mode

Using Interactive Mode

The following procedure lists the steps to install the schema for Oracle Real-Time Scheduler v2.3.0.2 using OraDBI.

Run the following command with the defined parameters on the command prompt from .. ORS\Install-Upgrade directory:

1. Open a command line prompt in the Windows environment.

2. Set Java Home.

   In the following example, JDK 1.8 is installed in C:\Program Files\Java\jdk1.8.0_101 directory.

   ```
   SET JAVA_HOME=C:\Program Files\Java\jdk1.8.0_101
   ```

3. Set the class path.

   In the following example, the required jarfiles, including OraDBI.jar, are available in the C:\InstallUpgrade\Jarfiles* directory.

   ```
   SET CLASSPATH=C:\InstallUpgrade\Jarfiles\*
   ```

4. Execute the following command:

   ```
   "%JAVA_HOME%"\bin\java -Xmx1500M -cp %CLASSPATH%
   com.oracle.ouaf.oem.install.OraDBI
   ```

   (or)

   ```
   "C:\Program Files\Java\jdk1.8.0_101"\bin\java -Xmx1500M -cp
   C:\InstallUpgrade\Jarfiles\* com.oracle.ouaf.oem.install.OraDBI
   ```
   The utility prompts you to enter values for the following parameters as per your environment:

   - •Name of the database server: <DB SERVER>

   - •Port no: <PORT>

   - •Name of the target database: <SID>

   - •Name of the owner of the database schema: <DBUSER>

   - •Password of the user name: <DBPASS>

   - •Location of Java Home: (e.g. C:\Java\jdk1.8.0): <Java Home>

   - •Oracle user with read-write privileges to the Database Schema: <CISUSER>

   - •Oracle user with read-only privileges to the Database Schema: <CISREAD>

   - •Oracle database role with read-write privileges to the Database Schema: <CIS_USER>

   - •Oracle database role with read-only privileges to the Database Schema: <CIS_READ>

   - •Enter the name of the target schema where you want to install or upgrade: <CISADM>

   - •Enter the password for the target schema: <CISADM password>

This process generates log files in the directory Install-Upgrade\logs. Make sure to check log files for any errors.

> **Note**: For OraDBI jar, you may receive the following message in the display output or logs. These errors can be safely ignored and the process should proceed to completion.

```
- 2016-05-23 16:31:38,315 [main] ERROR
(common.cryptography.KeyStoreWrapperFactory) The keystore file
'<filename>' does not exist.…
This file is either provided by the property
com.oracle.ouaf.system.keystore.file or expected to exist at the
```

```
default file location null Attempting to use the legacy
cryptography.
 - 2016-05-23 16:31:38,566 [main] INFO  (oem.install.OraDBI)
```

Oracle Real-Time Scheduler database component is now installed.

Using Command Line Mode
Run the following command with the defined parameters on the command prompt from
.. ORS\Install-Upgrade directory

```
"C:\Program Files\Java\jdk1.8.0_65\bin\java" -Xmx1500M -cp
C:\InstallUpgrade\Jarfiles\* com.oracle.ouaf.oem.install.OraDBI -d
jdbc:oracle:thin:@DB_Server:1521/SID,
DBUSER,DBPASS,RWUSER,RUSER,RW_USER_ROLE,R_USER_ROLE,DBUSER -l 1,2 -
j "C:\Program Files\Java\jdk1.8.0_65" C:\InstallUpgrade\Jarfiles -q
true
```

This process generates log files in the directory Install-Upgrade\logs. Make sure to check
log files for any errors.

> **Note**: For OraDBI jar, you may receive the following message in the display
> output or logs. These errors can be safely ignored and the process should
> proceed to completion.

```
- 2016-05-23 16:31:38,315 [main] ERROR
(common.cryptography.KeyStoreWrapperFactory) The keystore file
'<filename>' does not exist.…
This file is either provided by the property
com.oracle.ouaf.system.keystore.file or expected to exist at the
default file location null Attempting to use the legacy
cryptography.
 - 2016-05-23 16:31:38,566 [main] INFO  (oem.install.OraDBI)
```

Oracle Real-Time Scheduler database component is now installed.

Installing the CISADM Schema for Oracle Utilities Mobile Workforce Management

The files for the Initial Install of Oracle Utilities Mobile Workforce Management 2.3.0.2 are
located in the Install-Upgrade folder under MWM.

Follow the procedure below to install the database component of Oracle Utilities Mobile
Workforce Management.

Prepare the following parameters before installation:

- The name of the database server in which the database is configured- DB_SERVER

- The listener port number of the database - PORT

- The target database name in which the product is to be installed - SID

- A database user that will own the application schema (for example, CISADM) -
  DBUSER

- Password of the database user that will own the application schema - DBPASS

- A database user that has read-write (select/update/insert/delete) privileges to the objects
  in the application schema (for example, CISUSER). The application will access the
  database as this user - RWUSER

- A database user with read-only privileges to the objects in the application schema. (for
  example, CISREAD) - RUSER

- A database role that has read-write (select/update/insert/delete) privileges to the objects in the application schema. (for example, CIS_USER) - RW_USER_ROLE

- A database role with read-only privileges to the objects in the application schema. (for example, CIS_READ)- R_USER_ROLE

- Location for jar files. (The Jar files are bundled with the database package) - CLASSPATH

- Java Home (for example, C:\Java\jdk1.8.0) - JAVA_HOME

You can execute OraDBI.jar using either of the following methods:

- Using Interactive Mode

- Using Command Line Mode

Using Interactive Mode

The following procedure lists the steps to install the schema for Oracle Utilities Mobile Workforce Management v2.3.0.2 using OraDBI.

Run the following command with the defined parameters on the command prompt from .. MWM\Install-Upgrade directory:

1. Open a command line prompt in the Windows environment.

2. Set Java Home.

   In the following example, JDK 1.8 is installed in C:\Program Files\Java\jdk1.8.0_101 directory.

   ```
   SET JAVA_HOME=C:\Program Files\Java\jdk1.8.0_101
   ```

3. Set the class path.

   In the following example, the required jarfiles, including OraDBI.jar, are available in the C:\InstallUpgrade\Jarfiles* directory.

   ```
   SET CLASSPATH=C:\InstallUpgrade\Jarfiles\*
   ```

4. Execute the following command:

   ```
   "%JAVA_HOME%"\bin\java -Xmx1500M -cp %CLASSPATH%
   com.oracle.ouaf.oem.install.OraDBI
   ```

   (or)

   ```
   "C:\Program Files\Java\jdk1.8.0_101"\bin\java -Xmx1500M -cp
   C:\InstallUpgrade\Jarfiles\* com.oracle.ouaf.oem.install.OraDBI
   ```
   The utility prompts you to enter values for the following parameters as per your environment:

   •Name of the database server: <DB SERVER>

   •Port no: <PORT>

   •Name of the target database: <SID>

   •Name of the owner of the database schema: <DBUSER>

   •Password of the user name: <DBPASS>

   •Location of Java Home: (e.g. C:\Java\jdk1.8.0): <Java Home>

   •Oracle user with read-write privileges to the Database Schema: <CISUSER>

   •Oracle user with read-only privileges to the Database Schema: <CISREAD>

•Oracle database role with read-write privileges to the Database Schema: <CIS_USER>

•Oracle database role with read-only privileges to the Database Schema: <CIS_READ>

•Enter the name of the target schema where you want to install or upgrade: <CISADM>

•Enter the password for the target schema: <CISADM password>

This process generates log files in the directory Install-Upgrade\logs. Make sure to check log files for any errors.

> **Note**: For OraDBI jar, you may receive the following message in the display output or logs. These errors can be safely ignored and the process should proceed to completion.

```
-  2016-05-23 16:31:38,315 [main] ERROR
(common.cryptography.KeyStoreWrapperFactory) The keystore file
'<filename>' does not exist.…
This file is either provided by the property
com.oracle.ouaf.system.keystore.file or expected to exist at the
default file location null Attempting to use the legacy
cryptography.
 -  2016-05-23 16:31:38,566 [main] INFO  (oem.install.OraDBI)
```

Oracle Utilities Mobile Workforce Management database component is now installed.

Using Command Line Mode

Run the following command with the defined parameters on the command prompt from .. MWM\Install-Upgrade directory.

```
"C:\Program Files\Java\jdk1.8.0_65\bin\java" -Xmx1500M -cp
C:\InstallUpgrade\Jarfiles\* com.oracle.ouaf.oem.install.OraDBI -d
jdbc:oracle:thin:@DB_Server:1521/SID,
DBUSER,DBPASS,RWUSER,RUSER,RW_USER_ROLE,R_USER_ROLE,DBUSER -l 1,2 -
j "C:\Program Files\Java\jdk1.8.0_65" C:\InstallUpgrade\Jarfiles -q
true
```

This process generates log files in the directory Install-Upgrade\logs. Make sure to check log files for any errors.

> **Note**: For OraDBI jar, you may receive the following message in the display output or logs. These errors can be safely ignored and the process should proceed to completion.

```
-  2016-05-23 16:31:38,315 [main] ERROR
(common.cryptography.KeyStoreWrapperFactory) The keystore file
'<filename>' does not exist.…
This file is either provided by the property
com.oracle.ouaf.system.keystore.file or expected to exist at the
default file location null Attempting to use the legacy
cryptography.
 -  2016-05-23 16:31:38,566 [main] INFO  (oem.install.OraDBI)
```

Oracle Utilities Mobile Workforce Management database component is now installed.

ORADBI Performs the Following Tasks

• Interacts with the user to collect information about the name of Oracle account that will own the application schema (for example, CISADM), password of this account and the name of the Oracle account that the application user will use (for example, CISUSER), and the name of the Oracle account that will be assigned read-only privileges to the application schema (for example, CISREAD).

- Verifies whether tablespace names already exist in the Storage.xml file (if not, the process will abort).

- Verifies whether the upgrade path from the current release ID to the target release ID is supported by the upgrade.

- Upgrades the schema, installs the system data, and configures security.

- Maintains upgrade log tables in the database.

- Updates release ID when the upgrade is completed successfully.

- If an error occurs while executing a SQL script or another utility, it logs and displays the error message and allows you to re-execute the current step. Log files CdxDBI###.log are created in the same folder as CdxDBI and contains all the SQL commands executed against the database along with the results. The log files are incremental so that the results are never overwritten. If warning messages are generated during the upgrade, CdxDBI prompts the user at the end of the process. Users should check the log files to verify the warning messages.

- Warning messages are only alerts and do not necessary mean a problem exists.

- Stores the Schema owner and password in the feature configuration table. The password is stored in encrypted format.

Postinstallation Tasks
Importing Styles and Maps

Make sure that these steps are preformed from a Windows based machine.

To import styles and maps in the specified database, follow these steps:

1. Execute ImportSDOThemes.bat under the ORS/Post-Upgrade directory with the following parameters:

   ImportSDOThemes.bat -u CISADM -p <PASSWORD> -c <DATABASE NAME>

2. After importing themes/styles /GEOM Metadata (zz_user_sdo_themes/zz_user_sdo_styles, zz_user_sdo_metadata), postImport.sql is invoked from target main schema. This will:

   - Insert new styles/themes/sdo_geom_metadata in USER_SDO_THEMES/ USER_SDO_STYLES/USER_SDO_GEOM_METADATA

   - Drop ZZ_USER_SDO_THEMES, ZZ_USER_SDO_STYLES, ZZ_USER_SDO_GEOM_METADATA tables from target database.

   - Create domain key indexes

Enable USER_LOCK Package

For inbound web services to work the USER_LOCK must be enabled at the database level. This is a one-time step. If this is not already enabled please do so using the following steps.

1. Login as SYS user

2. On SQL prompt run:

   @?/rdbms/admin/userlock.sql

3. Grant permission by running the following SQL:

   grant execute on USER_LOCK to public;

Please note that grant can also be made to the database user which the Application connects to only instead of to public. For example, cisuser.

Populating Language Data

At this point, the Oracle Utilities Mobile Workforce Management database is ready for use. Please note that this database contains data in the ENGLISH language only. If you use any other supported language, run the F1-LANG batch program to duplicate the entries for new language

records. For more information on running this batch program, refer to the user documentation section "Defining Background Processes."

You can also install the language specific data packages (if available) into the database. Please contact your Oracle support representative to receive information on these packages.

Installing Service Packs and Patches

Periodically, Oracle Utilities releases a service pack of single fixes for products. A service pack is an update to an existing release that includes solutions to known problems and other product enhancements. A service pack is not a replacement for an installation, but a pack consisting of a collection of changes and additions for the installation. The service pack may include changes to be applied to the application server, the database, or both. The service pack includes all files necessary for installing the collection of changes, including installation instructions.

Between services packs, Oracle Utilities releases patches to fix individual bugs. For information on installing patches, see knowledge base article ID 974985.1 on My Oracle Support.

Service packs and patches can be downloaded from My Oracle Support (https://support.oracle.com/).

Demo Install

This section describes how to install the demo database components for Oracle Utilities Mobile Workforce Management, including:

- Copying and Decompressing Install Media
- Creating the Demo Database
- Postinstallation Tasks
- Installing Service Packs and Patches

Copying and Decompressing Install Media

To copy and decompress the Oracle Utilities Mobile Workforce Management database:

1. Download Oracle Utilities Mobile Workforce Management v2.3.0.2 from the Oracle Software Delivery Cloud.

2. Unzip the MWM-V2.3.0.2.0-database.zip files to a temporary folder. The Database folder contains several files that will be referred to during the installation process.

Creating the Demo Database

Note: You must have Oracle Database Server 12.1.0.1+ or Oracle Database Server 12.2.0.1+ installed on your machine in order to create the database.

For 12.2.0.1 set the compatible parameter to 12.1.0.2 before creating the database.

It is strongly recommended to use DBCA to create the database.

Creating the Database on UNIX

Create the database using the Database Configuration Assistant (DBCA).

Refer to https://docs.oracle.com/database/121/ADMIN/create.htm#ADMIN12479 for more information about 12c Release 1 database installation. Make sure to set character set for database as AL32UTF8.

Refer to Creating the Database for steps to create the database.

Note: Follow the Oracle database documentation for steps to create the database of the respective version.

Creating the Database on Windows

You should be logged in as a user who is a member of the local ORA_DBA group on that server. The ORA_DBA group should have "administrator" privileges assigned to it.

Refer to https://docs.oracle.com/database/121/ADMIN/create.htm#ADMIN12479 for more information about 12c Release 1 database installation. Make sure to set character set for database as AL32UTF8.

Refer to Creating the Database for steps to create the database.

Note: Follow the Oracle database documentation for steps to create the database of the respective version.

Importing the MWM Demo Dump File

After a successful database creation, demo data can also be imported by using by following these steps:

1. Set the correct ORACLE_SID and ORACLE_HOME.

2. Run following command to import the demo dump:

```
impdp directory= data_pump_dir dumpfile= expdp_demo.dmp
logfile=expdp_demo.log schemas=CISADM
```

> **Note**: The data_pump_dir must exist in the database created above before continuing with the import. You should also copy the expdp_demo.dmp file to the data_pump_dir. Decompress the expdp_demo.dmp.gz file first to extract the expdp_demo.dmp file. This file is in ..\MWM\Demo directory

Postinstallation Tasks

Populating Language Data

At this point, the Oracle Utilities Mobile Workforce Management demo database is ready for use. Please note that this database contains data in the ENGLISH language only. If you use any other supported language, you can run the F1-LANG batch program to duplicate the entries for new language records. For more information on running this batch program, refer to the user documentation section "Defining Background Processes."

You can also install the language specific demo data packages (if available) into the database. Please contact your Oracle representative to receive information on these packages.

Configuring Security

To configure security follow these steps:

1. Set PATH.

   In the following example, JDK 1.8 is installed in the C:\Program Files\Java\jdk1.8.0 directory.

   ```
   set PATH= C:\Program Files\Java\jdk1.8.0\bin;%PATH%
   ```

2. Set CLASSPATH.

   Copy the required jarfiles from the ..\DB\FW.V4.3.0.4.0\jarfiles folder, to the C:\Jarfiles directory.

   ```
   set CLASSPATH=C:\Jarfiles\*
   ```

3. Run the following command with the defined parameters on the command prompt.

   ```
   java com.oracle.ouaf.oem.install.OraGenSec -l oragensec.log -d
   <DBUSER>,<DBPASS>,jdbc:oracle:thin::@<DB_Server>:1521/
   <SERVICE_NAME> -a A -r <R_USER_ROLE>,<RW_USER_ROLE> -u
   <RWUSER>,<RUSER>
   ```

Installing Service Packs and Patches

Periodically, Oracle releases a service pack of single fixes for products. A service pack is an update to an existing release that includes solutions to known problems and other product enhancements. A service pack is not a replacement for an installation, but a pack consisting of a collection of changes and additions for the installation. The service pack may include changes to be applied to the application server, the database, or both. The service pack includes all files necessary for installing the collection of changes, including installation instructions.

Between services packs, Oracle releases patches to fix individual bugs. For information on installing patches, see knowledge base article ID 974985.1 on My Oracle Support.

Service packs and patches can be downloaded from My Oracle Support (https://support.oracle.com/).

Accelerator Install

This section describes how to install the Accelerator database components for Oracle Utilities Mobile Workforce Management including:

- Copying and Decompressing Install Media

- Creating the Accelerator Database

- Importing the MWM Accelerator Dump File

- Postinstallation Tasks

- Installing Service Packs and Patches

Copying and Decompressing Install Media

To copy and decompress the Oracle Utilities Mobile Workforce Management database:

1. Download Oracle Utilities Mobile Workforce Management v2.3.0.2 from the Oracle Software Delivery Cloud.

2. Unzip the MWM-V2.3.0.2-database.zip files to a temporary folder. The Database folder contains several files that will be referred to during the installation process.

Creating the Accelerator Database

> **Note**: You must have Oracle Database Server 12.1.0.1+ or Oracle Database Server 12.2.0.1+ installed on your machine in order to create the database. For 12.2.0.1 set the compatible parameter to 12.1.0.2 before creating the database.

It is strongly recommended that DBCA be used to create the database.

Creating the Database on UNIX

Create the database using the Database Configuration Assistant (DBCA).

Refer to https://docs.oracle.com/database/121/ADMIN/create.htm#ADMIN12479 for more information about 12c Release 1 database installation. Make sure to set character set for database as AL32UTF8.

> **Note**: Follow the Oracle database documentation for steps to create the database of respective version.

Refer to Creating the Database for steps to create the database.

Creating the Database on Windows

You should be logged in as a user who is a member of the local ORA_DBA group on that server. The ORA_DBA group should have "administrator" privileges assigned to it.

Refer to https://docs.oracle.com/database/121/ADMIN/create.htm#ADMIN12479 for more information about 12c Release 1 database installation. Make sure to set character set for database as AL32UTF8.

> **Note**: Follow the Oracle database documentation for steps to create the database of respective version.

Refer to Creating the Database for steps to create the database.

Importing the MWM Accelerator Dump File

After a successful database creation, Accelerator data can also be imported by following these steps:

1. Set the correct ORACLE_SID and ORACLE_HOME.

2. Run following command to import the Accelerator dump:

```
impdp directory= data_pump_dir dumpfile= expdp_acc.dmp
logfile=expdp_acc.log schemas=CISADM
```

Note: The data_pump_dir must exist in the database created above before continuing with the import. You should also copy the expdp_acc.dmp file to the data_pump_dir. Decompress the expdp_acc.dmp.gz file first to extract the expdp_acc.dmp file. This file is in the .\MWM\Accelerator directory.

Postinstallation Tasks

Populating Language Data

At this point, the Oracle Utilities Mobile Workforce Management Accelerator database is ready for use. Please note that this database contains data in the ENGLISH language only. If you use any other supported language, you can run the F1-LANG batch program to duplicate the entries for new language records. For more information on running this batch program, refer to the user documentation section "Defining Background Processes."

You can also install the language specific Accelerator data packages (if available) into the database. Please contact your Oracle representative to receive information on these packages.

Configuring Security

To configure security follow these steps:

1. Set PATH.

   In the following example, JDK 1.8 is installed in the C:\Program Files\Java\jdk1.8.0 directory.

   ```
   set PATH= C:\Program Files\Java\jdk1.8.0\bin;%PATH%
   ```

2. Set CLASSPATH.

   Copy the required jarfiles from the ..\DB\FW.V4.3.0.4.0\jarfiles folder, to the C:\Jarfiles directory.

   ```
   set CLASSPATH=C:\Jarfiles\*
   ```

3. Run the following command with the defined parameters on the command prompt.

   ```
   java com.oracle.ouaf.oem.install.OraGenSec -l oragensec.log -d
   <DBUSER>,<DBPASS>,jdbc:oracle:thin::@<DB_Server>:1521/
   <SERVICE_NAME> -a A -r <R_USER_ROLE>,<RW_USER_ROLE> -u
   <RWUSER>,<RUSER>
   ```

Installing Service Packs and Patches

Periodically, Oracle releases a service pack of single fixes for products. A service pack is an update to an existing release that includes solutions to known problems and other product enhancements. A service pack is not a replacement for an installation, but a pack consisting of a collection of changes and additions for the installation. The service pack may include changes to be applied to the application server, the database, or both. The service pack includes all files necessary for installing the collection of changes, including installation instructions.

Between services packs, Oracle releases patches to fix individual bugs. For information on installing patches, see knowledge base article ID 974985.1 on My Oracle Support.

Service packs and patches can be downloaded from My Oracle Support (https://support.oracle.com/).

Chapter 3

Database Design

This section provides a standard for database objects such as tables, columns, and indexes, for products using the Oracle Utilities Application Framework. This standard helps smooth integration and upgrade processes by ensuring clean database design, promoting communications, and reducing errors. Just as Oracle Utilities Application Framework goes through innovation in every release of the software, it is also inevitable that the product will take advantage of various database vendors' new features in each release. The recommendations in the database installation section include only the ones that have been proved by vigorous QA processes, field tests and benchmarks. This section includes:

- Database Object Standard
- Column Data Type and Constraints
- Standard Columns

Database Object Standard

This section discusses the rules applied to naming database objects and the attributes that are associated with these objects.

Categories of Data

A table can belong to one of the three categories:

- Control (admin)
- Master
- Transaction

For purposes of physical table space design, metadata and control tables can belong to the same category.

Example of tables in each category:

- **Control:** SC_USER, CI_ADJ_TYPE, F1_BUS_OBJ
- **Master:** CI_PER, CI_PREM,
- **Transaction:** F1_FACT, CI_FT

All tables have the category information in their index name. The second letter of the index carries this information. See "Indexes" on page 3 for more information.

Naming Standards

The following naming standards must be applied to database objects.

Table

Table names are prefixed with the owner flag value of the product. For customer modification **CM** must prefix the table name. The length of the table names must be less than or equal to 30 characters. A language table should be named by suffixing **_L** to the main table. The key table name should be named by suffixing **_K** to the main table.

It is recommended to start a table name with the 2-3 letter acronym of the subsystem name that the table belongs to. For example, **MD** stands for metadata subsystem and all metadata table names start with **CI_MD**.

Some examples are:

- CI_ADJ_TYPE
- CI_ADJ_TYPE_L

A language table stores language sensitive columns such as a description of a code. The primary key of a language table consists of the primary key of the code table plus language code (LANGAGUE_CD).

A key table accompanies a table with a surrogate key column. A key value is stored with the environment id that the key value resides in the key table.

The tables prior to V2.0.0 are prefixed with CI_ or SC_.

Columns

The length of a column name must be less than or equal to 30 characters. For customer modification, CM must prefix the column name. The following conventions apply when you define special types of columns in the database.

- Use the suffix **FLG** to define a lookup table field. Flag columns must be CHAR(4). Choose lookup field names carefully as these column names are defined in the lookup table (CI_LOOKUP_FLD) and must be prefixed by the product owner flag value.

- Use the suffix **CD** to define user-defined codes. User-defined codes are primarily found as the key column of the admin tables.

- Use the suffix **ID** to define system assigned key columns.

- Use the suffix **SW** to define Boolean columns. The valid values of the switches are 'Y' or 'N'. The switch columns must be CHAR(1)

- Use the suffix **DT** to define Date columns.

- Use the suffix **DTTM** to define Date Time columns.

- Use the suffix **TM** to define Time columns.

Some examples are:

- ADJ_STATUS_FLG

- CAN_RSN_CD

Indexes

Index names are composed of the following parts:

[OF][*application specific prefix*][C/M/T]NNN[P/S]n

- **OF**- Owner Flag. The standard is to use the two characters of the product's owner flag. Note that there may be some older indexes that use only the first character of the owner flag. For client specific implementation of index, use CM for Owner Flag. If implementation creates a CM Index on table-columns for which the base product already provides an index, then the CM Index will be overridden by the based index.

- Application specific prefix could be C, F, T or another letter.

- **C/M/T** - The second character can be either C or M or T. C is used for control tables (Admin tables). M is for the master tables. T is reserved for the transaction tables.

- **NNN** - A three-digit number that uniquely identifies the table on which the index is defined.

- **P/S** - P indicates that this index is the primary key index. S is used for indexes other than primary keys.

- **n** is the index number, unique across all indexes on a given table (0 for primary and 1, 2, etc., for the secondary indexes).

Some examples are:

- F1C066P0

- F1C066S1

- CMT206S2

    Warning! Do not use index names in the application as the names can change due to unforeseeable reasons.

Updating Storage.xml

The storage.xml file that comes with the product allocates all base tables and indexes to the default tablespace CISTS_01. If you decide to allocate some tables or indexes outside of the default tablespace, then this has to be reflected in the storage.xml file by changing the tablespace name from the default value to a custom value, according to the format shown below:

Format:

```
<Table_Name>
  <TABLESPACE>CISTS_01</TABLESPACE>
  <PARALLEL>1</PARALLEL>
- <LOB>
- <Column Name>
  <TABLESPACE>CISTS_01</TABLESPACE>
  <SECUREFILE>Y</SECUREFILE>
  <CHUNK>8192</CHUNK>
  <CACHE>N</CACHE>
  <LOGGING>Y</LOGGING>
  <INROW>Y</INROW>
  <COMPRESS>N</COMPRESS>
  </Column Name>
  </LOB>
  </Table_Name>
```

Where Parallel defines the number of threads, that Oracle DB Server will use to access a table or create an index.

We recommend creating CLOBs stored as SECUREFILE with Medium compression and Cache enabled. Please note that by default, medium compression is turned-off and must only be enabled if you have the Advanced compression license.

For instance, if a DBA decided to allocate table CI_ACCT in a tablespace MyTablespace, then they would have to change the storage.xml as follows:

```
<CI_ACCT>
<TABLESPACE>MyTablespace</TABLESPACE>
</CI_ACCT>
```

The oradbi process uses the storage.xml file to place the new database objects into defined tablespaces. A tablespace referenced in the storage.xml file must exist in the database.

The storage.xml file has to be adjusted before each upgrade and/or new installation as required to allocate the tables and indexes across those tablespaces.

Table name is included as a comment for each of the indexes for clarity.

For initial installs, information for each object should be reviewed by a DBA. For upgrades, only tablespace information for the objects added in the new release needs to be reviewed by a DBA.

Be careful while editing this file. Make sure that the tablespace names being used exist in the database. Do not change the basic format of this file.

Sequence

The base sequence name must be prefixed with the owner flag value of the product. For customer modification **CM** must prefix the sequence name. The sequence numbers should be named as below:

1. If the Sequence is used for a specific table, then use the following sequence name:

    [OF][C/M/T]NNN_SEQ

- OF stands for Owner Flag. For example, for Framework its F1. Other examples are M1 etc.

- C/M/T stands for Control (Admin)/Master/Transaction Tables.

- NNN is a three digit unique Identifier for a table on which the sequence is defined.

    For e.g: F1T220_SEQ

2. If more than one sequence is used for a specific table, then use the following Sequence Name:

[OF][C/M/T]NNN_Column_Name_SEQ

- OF stands for Owner Flag. For example, for framework is F1. Other examples are M1 etc.

- C/M/T stands for Control (Admin)/Master/Transaction tables.

- NNN is a three digit unique identifier for a table on which the sequence is defined.

  For Example: F1T220_BO_STATUS_CD_SEQ and F1T220_BUS_OBJ_CD_SEQ

3. If sequence is used for a generic requirement and not specific to a table, then use the following sequence name.

   [OF]Column_Name_SEQ

- OF stands for Owner Flag. For example, for framework is F1. Other examples are M1 etc.

  For Example: F1FKVALID_SEQ

- For a customer modification, CM must prefix the sequence name.

Trigger

The base trigger name must be prefixed with the owner flag value of the product.

When implementers add database objects, such as tables, triggers and sequences, the name of the objects should be prefixed by CM.

Column Data Type and Constraints

This section discusses the rules applied to column data type and constraints, and the attributes that are associated with these objects.

User Defined Code

User Defined Codes are defined as CHAR type. The length can vary by the business requirements but a minimum of eight characters is recommended. You will find columns defined in less than eight characters but with internationalization in mind, new columns should be defined as CHAR(10) or CHAR(12). Also note that when the code is referenced in the application the descriptions are shown to users in most cases.

System Assigned Identifier

System assigned random numbers are defined as CHAR type. The length of the column varies to meet the business requirements. Number type key columns are used when a sequential key assignment is allowed or number type is required to interface with external software. For example, Notification Upload Staging ID is a Number type because most EDI software uses a sequential key assignment mechanism. For sequential key assignment implementation, the DBMS sequence generator is used in conjunction with Number Type ID columns.

Date/Time/Timestamp

Date, Time and Timestamp columns are defined physically as DATE in Oracle. Non-null constraints are implemented only for the required columns.

Number

Numeric columns are implemented as NUMBER type in Oracle. The precision of the number should always be defined. The scale of the number might be defined. Non-null constraints are implemented for all number columns.

Fixed Length/Variable Length Character Columns

When a character column is a part of the primary key of a table define the column in CHAR type. For the non-key character columns, the length should be the defining factor. If the column length should be greater than 10, use VARCHAR2 type in Oracle.

Null Column Support

The product supports Nullable columns. This means that the application can write NULLs instead of a blank space or zero (for numeric columns) by using NULLABLE_SW on CI_MD_TBL_FLD. If REQUIRED_SW is set to 'N' and the NULLABLE_SW is set to 'Y', the application will write a NULL in that column. The artifact generator will create hibernate mapping files with appropriate parameters so that the framework hibernate mapping types will know if a given property supports a null value.

NULLABLE_SW is not new, but has previously been used for certain fields such as dates, and some string and number foreign-key columns. Because of this, there is the possibility that there is incorrect metadata for some columns, and that turning on this new feature could result in incorrect behavior when using that metadata. The upgrade script fixes the metadata to make sure that the existing tables will not be affected.

This new feature only supports tables maintained by Java but NOT a Java program converted from COBOL. Thus, enhancing any existing tables to use null columns must be done only after making sure that the tables are maintained by Java, and not Java converted COBOL programs.

XML Type Support

The product supports XML Type. XML Type provides following advantages

1. The ability to use XQuery for querying nodes in the XML document stored within a column defined as XMLType.

2. The option to use the XML engine, which is built into the Oracle Database, to create indexes using nodes within the XML document stored in the XMLType column.

Cache and Key Validation Flags

By default, the Cache Flag is set to NONE. For most of the admin tables the CACHE Flag should be 'Cached for Batch'. This specifies that the table is cached as L2 cache to reduce database trips.

By default the Key Validation Flag is set to ALL. For tables which have the user defined keys, the KEY_VALIDATION_FLG should be set as 'ALL'. This checks the existence of the key before inserting a new one.

Default Value Setting

The rules for setting the database default values are as follows:

• When a predefined default value is not available, set the default value of Non-null CHAR or VARCHAR columns to blank except the primary key columns.

• When a predefined default value is not available, set the default value Non-null Number columns to 0 (zero) except the primary key columns.

• No database default values should be assigned to the Non Null Date, Time, and Timestamp columns.

Foreign Key Constraints

Referential integrity is enforced by the application. In the database do not define FK constraints. Indexes are created on most of Foreign Key columns to increase performance.

Standard Columns

This section discusses the rules applied to standard columns and the attributes that are associated with these objects.

Owner Flag

Owner Flag (OWNER_FLG) columns exist on the system tables that are shared by multiple products. Oracle Utilities Application Framework limits the data modification of the tables that have owner flag to the data owned by the product.

Version

The Version column is used to for optimistic concurrency control in the application code. Add the Version column to all tables that are maintained by a Row Maintenance program.

Chapter 4

Database Implementation Guidelines

The following section outlines the general implementation guidelines for the database components, including:

- Configuration Guidelines
- Oracle Database Implementation Guidelines.

> **Note**: Refer to My Oracle Support for more information.

Configuration Guidelines

This section includes general recommendations for configuring various database objects and includes a brief syntax overview. It covers the general aspects of the database objects and does not cover any specific implementation requirements. This section includes:

- Index
- Table Partitioning Recommendations
- Transparent Data Encryption Recommendations
- Data Compression Recommendations
- Database Vault Recommendations
- Oracle Fuzzy Search Support
- Information Lifecycle Management (ILM) and Data Archiving Support
- Storage Recommendations
- Database Configuration Recommendations
- Database Syntax
- Database Initialization Parameters

Index

Index recommendations specify points that need to be considered when creating indexes on a table.

1. Indexes on a table should be created according to the functional requirements of the table and not in order to perform SQL tuning.

2. The foreign keys on a table should be indexes.

   Note: If the implementation creates a CM index on table-columns where the product already provides an index, then the CM index will be overridden by the base index.

Table Partitioning Recommendations

Oracle Utilities recommends using a minimum of 'n' partitions for selective database objects, where 'n' is number of RAC nodes.

Transparent Data Encryption Recommendations

Oracle Utilities supports Oracle Transparent Data Encryption (TDE). Oracle 11gR1 supports tablespace level encryption. The application supports tablespace level encryption for all application data. Make sure that the hardware resources are sufficiently sized for this as TDE uses additional hardware resources. The Oracle Advanced Security license is a prerequisite for using TDE.

Please consider the following when implementing TDE:

- Create a wallet folder to store the master key. By default, the wallet folder should be created under $ORACLE_BASE/admin/<sid>.

- The wallet containing the master key can be created using the following command:

  ```
  alter system set encryption key authenticated by "keypasswd"
  ```

- The wallet can be closed or opened using the following commands:

```
alter system set wallet open identified by "keypasswd";
alter system set wallet close;
```

- Column level encryption can be achieved using the following commands:

```
create table <table_name>
(name varchar2(200) default ' ' not null,
bo_data_area CLOB encrypt using 'AES128',
bo_status_cd char(12)    encrypt using 'AES128')
lob (bo_data_area) store as securefile (cache compress)
tablespace <tablespace_name>;
```

- AES128 is the default encryption algorithm.

- Tablespace level encryption is also supported using the following command:

```
Create tablespace <tablespace_name> logging datafile '<datafile
location>' size <initial size> reuse autoextend on next <next
size>
maxsize unlimited extent management local uniform size
<uniform size> encryption using 'AES128' default
storage(encrypt);
```

- Indexed columns can only be encrypted using the NO SALT Option. Salt is a way to strengthen the security of encrypted data. It is a random string added to the data before it is encrypted, causing repetition of text in the clear to appear different when encrypted.

Data Compression Recommendations

Oracle Utilities supports Advanced Data Compression, available with Oracle 11gR1 onwards, to reduce the database storage footprint. Make sure that your resources are sufficiently sized for this as it uses additional system resources. Compression can be enabled at the Tablespace level or at the Table level.

Exadata Hardware

For Exadata hardware the compression recommendations are:

- For high volume tables, keep the current table partition uncompressed. All of the older partitions will be compressed based on QUERY HIGH compression.

- For high volume tables with CLOBs, always keep the CLOBs in securefiles with MEDIUM compression. Also keep the current table partition uncompressed. All of the older partitions will be compressed based on QUERY HIGH compression.

- Load data into the uncompressed table partitions using a conventional load and then, once data is loaded using a CTAS operation, load into a temporary heap table. Then truncate the original partition. Alter the original partition into HCC compressed and then partition exchange this with the temporary heap table.

- All multi column Indexes (primary as well as secondary) will be compressed using the default compression. HCC or OLTP compression is not applicable on the top of compressed Indexes.

Non- Exadata Hardware

For non-Exadata hardware the recommendations are the same as above, except that you cannot use HCC compression (it is only available in Exadata database machine). Instead of HCC you can use any other compression tool available to you for non-Exadata hardware.

CLOB Fields

All CLOB fields should be stored as SecureFiles and Medium compressed. This requires a separate license for Advanced Data Compression. As a part of the schema, we create the product-owned tables with compression turned OFF at the LOB level. If you have the license for Advanced Data Compression, you can enable compression by updating the storage.xml.

Compression Guidelines

- Admin and Metadata tables and their indexes will NOT be compressed.

- All Transactional Tables will be compressed.
  This includes ILM enabled MOs where applicable.

- Compression will be done at the tablespace level.

- Different MOs will have different tablespaces.

- Partitioned MOs will have one tablespace per partition.

- Child tables will use reference partitioning with parent + children sharing the same tablespace. (parent and child will always be managed/archived together).

- All multicolumn indexes on transactional/ILM tables will be compressed.

- Use 'compress advanced low'.

- Local partitioned indexes will reside in the same tablespace as the table.

- Each MO will have an index tablespace. All MO (Parent-Child Table) indexes will share this tablespace.

- Do NOT specify standard index compression.

- Securefile medium compression in row for LOBs and CLOBs.

Database Vault Recommendations

The product supports Database Vault. All non-application User IDs can be prevented from using DDL or DML statements against the application schema. So SYS and SYSTEM cannot issue DDL or DML statements against CISADM schema.

The application-specific administration account can issue DDL statements but should not be able to perform any DML or DCL statements.

Application user must be given DML only permissions.

Database Vault can be used to control access during patch process and Install/Upgrade process.

Oracle Fuzzy Search Support

The product supports Oracle Fuzzy searches. To use this feature, Oracle Text must be installed. After Oracle Text is installed, an index must be created on the table where the fuzzy search needs to be performed from the application. This is only an Oracle database option and is not supported by other databases. Additionally, not all languages are supported. Refer to the Oracle database documentation for more information about fuzzy searching.

A typical syntax for implementation of fuzzy searching is as below. For the most updated syntax, please refer to Oracle Fuzzy documentation.

```
GRANT CTXAPP TO <Application schema owner e.g CISADM>;

GRANT EXECUTE ON CTX_DDL TO <Application schema owner e.g CISADM>;

create index <Application schema owner e.g CISADM>.<Index_Name> on
Application schema owner e.g CISADM>.<Table_Name> (<column_name>)
indextype is ctxsys.context parameters ('sync (on commit)');
begin
ctx_ddl.sync_index('Application schema owner e.g
CISADM>.<Index_Name>');
end
/
```

Information Lifecycle Management (ILM) and Data Archiving Support

The product supports Data Archiving based on Information Lifecycle Management (ILM). If Information Lifecycle Management is part of your implementation, please refer to the chapter Information Lifecycle Management and Data Archiving in this guide for instructions on partitioning objects when using ILM.

Storage Recommendations

This section specifies recommended options for storing the database objects.

SecureFile for Storing LOBs

Beginning with Oracle 11g, tables having fields with data type of CLOB or BLOBS should have the LOB Columns stored as SecureFiles.

- The storage options with SecureFiles for Heap Tables should be ENABLE STORAGE IN ROW, CACHE and COMPRESS.

- For the IOT Table the PCTTHRESHOLD 50 OVERFLOW clause should be specified and the storage options with SecureFiles should be ENABLE STORAGE IN ROW, CACHE and COMPRESS.

- The PCTTHRESHOLD should be specified as a percentage of the block size. This value defines the maximum size of the portion of the row that is stored in the Index block when an overflow segment is used.

- The CHUNK option for storage, which is the data size used when accessing or modifying LOB values, can be set to higher than one database block size if big LOBs are used in the IO Operation.

- For SecureFiles, make sure that the initialization parameter db_securefile is set to ALWAYS.

- The Tablespace where you are creating the SecureFiles should be enabled with Automatic Segment Space Management (ASSM). In Oracle Database 11g, the default mode of Tablespace creation is ASSM so it may already be set for the Tablespace. If it's not, then you have to create the SecureFiles on a new ASSM Tablespace.

  Note: To enable compression on SecureFiles, you must have an Oracle Advanced Compression license in addition to Oracle Database Enterprise Edition. This feature is not available for the standard edition of the Oracle database.

If you are using Oracle Database Enterprise Edition, please verify that the "COMPRESS" flag is turned on by setting it to "Y" in Storage.xml.

Refer to the Database Syntax section for more information on SecureFiles.

Database Configuration Recommendations

This section specifies the recommended methods for configuring the database with a focus on specific functional area.

Large Redo Log File Sizes

The Redo Log files are written by the Log Writer Background process. These log files are written in a serial manner. Once a log File is full, a log switch occurs and the next log file starts getting populated.

It is recommended that the size of the Redo log files should be sufficiently high so that you do not see frequent Log Switches in the alert logs of the database. Frequent Log Switches impact the IO performance and can be avoided by having a larger Redo log file size.

Frequent Log Switches impacts the IO performance and can be avoided by having a bigger Redo log File Size.

Database Syntax

SecureFile

```
CREATE TABLE <Table_Name>
     ( COLUMN1 ….,
       COLUMN2 (CLOB)
     )
LOB(COLUMN2) STORE AS SECUREFILE (CACHE COMPRESS);

CREATE TABLE <Table_Name>
     ( COLUMN1 ….,
       COLUMN2 (CLOB)
       CONTRAINT <> PRIMARY KEY(…)
     )
ORGANIZATION INDEX PCTTHRESHOLD 50 OVERFLOW
LOB(COLUMN2) STORE AS SECUREFILE (ENABLE STORAGE IN ROW CHUNK CACHE
COMPRESS);
```

Database Initialization Parameters

The recommended initialization parameters are given below. These parameters are a starting point for database tuning. An optimal value for a production environment may differ from one customer deployment to another.

db_block_size=8192

log_checkpoint_interval=0

db_file_multiblock_read_count=8

transactions=3000

open_cursors=30000

db_writer_processes=10

db_files=1024

dbwr_io_slaves=10   (Only if Asynchronous IO is not Supported)

sessions=4500

memory_target=0

memory_max_target=0

processes=3000

dml_locks=48600

_b_tree_bitmap_plans=FALSE

Oracle Database Implementation Guidelines

This section provides specific guidelines for implementing the Oracle database.

Oracle Partitioning

If you use a base index for the partitioning key, rename the index to CM**.

If you use the primary key index of the table as the partitioning key:

- Make the index non-unique.

- Primary constraints should still exist.

The upgrade on the partitioned table will work the best if the partitioning key is not unique. This allows the upgrade tool to drop the PK constraints if the primary key columns are modified and recreate the PK constraints without dropping the index.

Database Statistics

During an install process new database objects may be added to the target database. Before starting to use the database, generate the complete statistics for these new objects by using the DBMS_STATS package. You should gather statistics periodically for objects where the statistics become stale over time because of changing data volumes or changes in column values. New statistics should be gathered after a schema object's data or structure are modified in ways that make the previous statistics inaccurate.

For example, after loading a significant number of rows into a table, collect new statistics on the number of rows. After updating data in a table, you do not need to collect new statistics on the number of rows, but you might need new statistics on the average row length.

A sample syntax that can be used is as following.

```
BEGIN
SYS.DBMS_STATS.GATHER_SCHEMA_STATS (
OwnName => 'CISADM'
,Degree => 16
,Cascade => TRUE
,Method_opt => 'FOR ALL COLUMNS SIZE AUTO'
,Granularity => 'ALL');
END;
/
```

Chapter 5

Information Lifecycle Management and Data Archiving

Oracle Utilities Mobile Workforce Management provides support for Data Archiving based on Information Lifecycle Management (ILM).

ILM is methodology designed to address data management issues, with a combination of processes, policies, software and hardware so that the appropriate technology can be used for each phase of the lifecycle of the data. The lifecycle of data typically refers to the fact that the most recent data is active in the system and as time goes the same data becomes older and older. Older data starts becoming an overhead to the overall application not only in terms of storage but also in terms of performance. This older data can be compressed using advanced compression techniques like Advanced Compression and Hybrid columnar compression, and can be put into slower and cheaper storage media and can be possibly removed from the system to make an overall savings in terms of cost and performance.

This chapter includes:

- ILM Implementation Overview
- ILM Implementation Components
- ILM Database Administrator's Tasks

ILM Implementation Overview

The implementation of ILM for products based on Oracle Utilities Application Framework includes a combination of configuration in the application and configuration in the database.

The approach followed for most maintenance objects that support ILM is the concept that the age of the data is not the only criterion used to determine the lifecycle of a record. There may be business rules that dictate that some "old" records are still valid and should not be archived yet.

The Oracle Utilities Application Framework includes the combination of an ILM date and an archive switch to indicate if the record can be archived or not. The switch is set using processes that periodically review "old" data and call an algorithm to assess whether or not the record can be archived.

Database Administrators can then review older partitions and review the number of records that have the archive switch set to Y. If this is set on all records, the DBA is safe to take desired steps for that partition. If there are some records that are still not eligible to be archived, the DBA may choose to move those records to a different partition to free up the partition for desired steps.

ILM Implementation Components

The ILM based solution contains a number of components.

- ILM Specific Table Columns - For any Maintenance Object (MO) that has been configured to support ILM, the primary table of the MO includes two columns: ILM Date and ILM Archive switch.

  - ILM_DT - This field is the date used for ILM. It is defaulted to an appropriate date (typically the system date) when a new record is inserted. This column should be rarely updated. This will make sure that during the online/batch processing there is no extra overhead of moving these records from one partition to another. In circumstances where the record is not eligible for archiving, an algorithm (or a DBA) can update the ILM Date to the current system date (or another appropriate "newer" date) to remove it from a partition that is ready to be archived.

  - ILM_ARCHIVE_SW - This field is set to N (Not yet eligible for archiving) when a new record is inserted. Subsequent reviews of "old" records may assess the data and change the value to "Y" based on business rules.

- Referential Integrity Constraints- To use ILM within the database, database level foreign key constraints must be added to child tables of ILM enabled MOs

- Partitioning - Partitioning is mandatory for ILM implementation. This provides segregation of data into multiple table partitions which provides better management of the data over its lifecycle.

ILM Database Administrator's Tasks

For a database administrator, there are two key phases involved with managing your data using ILM.

- Preparation Phase - This phase covers the database level configuration that needs to be done before the ILM solution runs in a production environment.

- On-going Maintenance Phase - This phase covers the ongoing maintenance tasks.

Preparation Phase

The following steps provide a high-level overview of the steps that need to be performed to implement ILM on enabled MOs.

Please refer to Appendix E: Sample Statements and DDL for ILM Enabled Maintenance Objects for detailed information using To Do Entry as an example.

1. Backup the existing tables, and primary key index associated with ILM enabled MOs by renaming the tables.

2. Save the DDLs for the secondary indexes as you will need to recreate them later.

3. Drop secondary indexes on the renamed tables.

4. Create Partitioned table with no secondary indexes for ILM enabled MOs using a CTAS operation (Create Table as Select), which will also load the data into the partitioned table structure.

   Functional Note: ILM enabled MOs should have the ILM date (ILM_DT) populated when data is moved into the new partitioned table. Please refer to the Module Specific ILM Implementation Details section below for initial load details on which date column to use as the basis for populating the ILM date. Often it is based on Create Date (CRE_DTTM). ILM_ARCH_SW should initially be set to 'N'.

5. Enable logging option.

6. Create Primary index.

7. Create Primary key of parent table.

8. Create secondary indexes for the newly-created partitioned tables. This includes creating an index used specifically to benefit the ILM Crawler batch. The recommendation for this index name is to prefix it with "ILM".

   Note: This can be created specifying parallel index create; remember to turn off parallelism after the index is created.

9. Follow similar operation for all child tables for this MO, such as rename child table, and primary key index, generate DDL for secondary index, drop secondary index etc. Sample DDL for child tables, their partitioning and indexes can be found in Appendix E: Sample Statements and DDL for ILM Enabled Maintenance Objects.

10. Drop the backup tables after verifying the newly created partitioned tables.

11. Rename the following index:

```
ALTER INDEX XT039S8  RENAME TO ILM_XT039S8;
```

Module Specific ILM Implementation Details

This section outlines each maintenance object that has been configured to support ILM. The parent table is noted. Other tables are child tables of the parent unless otherwise noted. In each case, the partitioning strategy is indicated only if it differs from the general recommendation described above.

Each table's indexes are listed with a recommendation of whether the index should be built as a global or local index and whether the index should be partitioned. In addition to the base delivered indexes, each primary (parent) table includes a recommended ILM specific local index to build with the ILM_DT, ILM_ARCH_SW and the primary key of the table. In each addition, the recommendation of the initial load of the ILM_DT for existing records is noted.

This section details the following maintenance objects:

- To Do Entry
- Sync Request (Outbound)
- Inbound Sync Request
- Outbound Message
- Service Task
- Object Revision
- Crew Shift
- Task
- Period of Unavailability
- Remote Message
- Alert
- Dispatcher Shift
- Mail
- Procedure
- Timesheet
- GPS Data

To Do Entry

This table describes the To Do Entry maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
CI_TD_ENTRY (Parent)	RANGE (ILM_DT, TD_ENTRY_ ID)					CI_TD_ENTRY. CRE_DTTM
		XT039P0	TD_ENTRY_ID	Global Partitioned	RANGE (TD_ENTRY_ID)	
		XT039S2	ASSIGNED_TO, TD_ENTRY_ID	Global		
		XT039S3	ENTRY_STATUS_FLG, ASSIGNED_TO	Global		
		XT039S4	ROLE_ID, TD_TYPE_CD, ENTRY_STATUS_FLG, TD_PRIORITY_FLG	Global		
		XT039S5	BATCH_CD, BATCH_NBR, ENTRY_STATUS_FLG	Global		
		XT039S6	TD_ENTRY_ID, ASSIGNED_TO, ENTRY_STATUS_FLG	Global		
		XT039S7	COMPLETE_USER_ID, COMPLETE_DTTM, TD_ENTRY_ID	Global		
		ILM_XT0 39S8	ILM_DT, ILM_ARCH_SW, TD_ENTRY_ID	Local Partitioned		
CI_TD_ENTRY_ CHA	Reference Partitioning	XT701P0	TD_ENTRY_ID, CHAR_TYPE_CD, SEQ_NUM	Global Partitioned		
		XT701S1	SRCH_CHAR_VAL, CHAR_TYPE_CD, TD_ENTRY_ID	Global		
CI_TD_DRLKEY	Reference Partitioning	XT037P0	TD_ENTRY_ID, SEQ_NUM	Global Partitioned		
		XT037S1	KEY_VALUE, TD_ENTRY_ID	Global		
CI_TD_LOG	Reference Partitioning	XT721P0	TD_ENTRY_ID, SEQ_NUM	Global Partitioned		
		XT721S1	LOG_DTTM,USER_ID, LOG_TYPE_FLG, TD_ENTRY_ID	Global		

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
CI_TD_MSG_PARM(Child table of CI_TD_LOG)	Reference Partitioning	XT040P0	TD_ENTRY_ID, SEQ_NUM	Global		
CI_TD_SRTKEY	Reference Partitioning	XT041P0	TD_ENTRY_ID, SEQ_NUM	Global Partitioned		
		XT041S1	KEY_VALUE, TD_ENTRY_ID	Global		

Sync Request (Outbound)

This table describes the Sync Request (Outbound) maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
F1_SYNC_REQ (Parent)	RANGE (ILM_DT, F1_SYNC_REQ_ID)				RANGE (F1_SYNC_REQ_ID)	F1_SYNC_REQ.CRE_DTTM
		F1T014P0	F1_SYNC_REQ_ID	Global Partitioned		
		F1T014S1	BO_STATUS_CD, BUS_OBJ_CD, F1_SYNC_REQ_ID	Global		
		F1T014S2	BO_STATUS_REASON_CD	Global		
		F1T014S3	MAINT_OBJ_CD, PK_VALUE1, PK_VALUE2, F1_SYNC_REQ_ID	Global		
		ILM_F1T014S4	ILM_DT, ILM_ARC_SW, F1_SYNC_REQ_ID	Local Partitioned		

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
F1_SYNC_REQ_CHAR	Reference Partitioning	F1T017P0	F1_SYNC_REQ_ID, CHAR_TYPE_CD, SEQ_NUM	Global Partitioned		
		F1T017S1	SRCH_CHAR_VAL	Global		
F1_SYNC_REQ_EXTRACT	Reference Partitioning	F1T019P0	F1_SYNC_REQ_ID, SEQ_NUM	Global Partitioned		
F1_SYNC_REQ_LOG	Reference Partitioning	F1T015P0	F1_SYNC_REQ_ID, SEQNO	Global Partitioned		
		F1T015S1	CHAR_TYPE_CD, CHAR_VAL_FK1	Global		
		F1T015S2	CHAR_TYPE_CD, CHAR_VAL	Global		
		F1T015S3	BO_STATUS_REASON_CD	Global		
F1_SYNC_REQ_LOG_PARM (Child Table of F1_SYNC_REQ_LOG_PARM)	Reference Partitioning	F1T016P0	F1_SYNC_REQ_ID, SEQNO, PARM_SEQ	Global Partitioned		

Inbound Sync Request

This table describes the Inbound Sync Request maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
F1_SYNC_REQ_IN (Parent)	RANGE(ILM_DT, F1_SYNC_REQ_IN_ID)				RANGE (F1_SYNC_REQ_IN_ID)	F1_SYNC_REQ_IN.CRE_DTTM
		F1T191P0	F1_SYNC_REQ_IN_ID	Global Partitioned		
		F1T191S1	BO_STATUS_CD, BUS_OBJ_CD, F1_SYNC_REQ_IN_ID	Global		

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
		F1T191S2	MAINT_OBJ_CD, EXT_PK_VALUE1, NT_XID_CD, PK_VALUE1	Global		
		ILM_F1T191S3	ILM_DT, ILM_ARCH_SW, F1_SYNC_REQ_IN_ID	Local Partitioned		
F1_SYNC_REQ_IN_CHAR	Reference Partitioning	F1T193P0	F1_SYNC_REQ_IN_ID, CHAR_TYPE_CD, SEQ_NUM	Global Partitioned		
		F1T193S1	SRCH_CHAR_VAL	Global		
F1_SYNC_REQ_IN_EXCP	Reference Partitioning	F1T197P0	F1_SYNC_REQ_IN_ID, SEQNO	Global Partitioned		
F1_SYNC_REQ_IN_EXCP_PARM (Child Table of F1_SYNC_REQ_IN_EXCP)	Reference Partitioning	F1T198P0	F1_SYNC_REQ_IN_ID, SEQNO, PARM_SEQ	Global Partitioned		
F1_SYNC_REQ_IN_LOG	Reference Partitioning	F1T194P0	F1_SYNC_REQ_IN_ID, SEQNO	Global Partitioned		
		F1T194S1	CHAR_TYPE_CD, CHAR_VAL_FK1	Global		
		F1T194S2	CHAR_TYPE_CD, CHAR_VAL	Global		
F1_SYNC_REQ_IN_LOG_PARM (Child Table of F1_SYNC_REQ_IN_LOG)	Reference Partitioning	F1T195P0	F1_SYNC_REQ_IN_ID, SEQNO, PARM_SEQ	Global Partitioned		
F1_SYNC_REQ_IN_REL_OBJ	Reference Partitioning	F1T192P0	F1_SYNC_REQ_IN_ID, MAINT_OBJ_CD, REL_OBJ_TYPE_FLG	Global Partitioned		

Outbound Message

This table describes the Outbound Message maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
F1_OUTMSG (Parent)	RANGE (ILM_DT, OUTMSG_ID)				RANGE (OUMSG_ID)	F1_OUTMSG. CRE_DTTM
		FT010P0	OUTMSG_ID	Global Partitioned		
		FT010S1	OUTMSG_STATUS_FLG, OUTMSG_TYPE_CD	Global		
		ILM_FT010S2	ILM_DT, ILM_ARC_SW, OUTMSG_ID	Local Partitioned		
F1_OUTMSG_ERRPARM	Reference Partitioning	FT011P0	OUTMSG_ID, PARM_SEQ	Global Partitioned		

Service Task

This table describes the Service Task maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
F1_SVC_TASK (Parent)	RANGE (ILM_DT, F1_SVC_TASK_ID)				RANGE (F1_SVC_TASK_ID_)	F1_SVC_TASK. CRE_DTTM
		F1C474P0	F1_SVC_TASK_ID	Global Partitioned		
		F1C474S1	F1_STASK_TYPE_CD	Global		
		F1C474S2	BUS_OBJ_CD	Global		
		ILM_F1C474S2	ILM_DT, ILM_ARC_SW, F1_SVC_TASK_ID	Local Partitioned		
F1_SVC_TASK_CHAR	Reference Partitioning	F1C476P0	F1_SVC_TASK_ID, CHAR_TYPE_CD, SEQ_NUM	Global Partitioned		

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
		F1C476S1	SRCH_CHAR_VAL	Global		
F1_SVC_TASK_ LOG	Reference Partitioning	F1C477P0	F1_SVC_TASK_ID, SEQNO	Global Partitioned		
		F1C477S1	CHAR_TYPE_CD, CHAR_VAL_FK1	Global		
		F1C477S2	CHAR_TYPE_CD, CHAR_VAL	Global		
F1_SVC_TASK_ LOG_PARM (Child Table of F1_SVC_TASK_ LOG)	Reference Partitioning	F1C478P0	F1_SVC_TASK_ID, SEQNO, PARM_SEQ	Global Partitioned		
F1_SVC_TASK_ REL_OBJ	Reference Partitioning	F1C479P0	F1_SVC_TASK_ID, MAINT_OBJ_CD, SEQ_NUM	Global Partitioned		
		F1C479S1	MAINT_OBJ_CD, PK_VALUE1, PK_VALUE2, PK_VALUE3	Global		

Object Revision

This table describes the Object Revision maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
F1_OBJ_REV (Parent)	RANGE (ILM_DT, REV_ID)				RANGE (REV_ID)	F1_OBJ_REV. STATUS_UPD_D TTM
		FT035P0	REV_ID	Global Partitioned		

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
		FT035S1	BO_STATUS_CD, BUS_OBJ_CD, REV_ID	Global		
		FT035S2	MAINT_OBJ_CD, PK_VALUE1	Global		
		FT035S3	EXT_REFERENCE _ID, MAINT_OBJ_CD	Global		
		FT035S4	USER_ID, MAINT_OBJ_CD	Global		
		FT035S5	PK_VALUE1	Global		
		ILM_FT035S6	ILM_DT, ILM_ARC_SW, REV_ID	Local Partitioned		
F1_OBJ_REV_ CHAR	Reference Partitioning	FT037P0	REV_ID, CHAR_TYPE_CD, SEQ_NUM	Global Partitioned		
		FT037S1	SRCH_CHAR_VAL	Global		
F1_OBJ_REV_ LOG	Reference Partitioning	FT039P0	REV_ID, SEQNO	Global Partitioned		
F1_OBJ_REV_ LOG_PARM (Child Table of F1_OBJ_REV_ LOG)	Reference Partitioning	FT040P0	REV_ID, SEQNO, PARM_SEQ	Global Partitioned		

Note: This maintenance object is enabled for ILM, however it is not used in a production environment. It is typically used in a development or configuration environment. Your implementation should review its use of this functionality and consider whether or not it is a candidate for ILM and in which region.

Crew Shift

This table describes the Crew Shift maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
M1_CREW_SHFT (Parent)	RANGE (ILM_DT, CREW_SHFT_ID)	MT100P0	CREW_SHFT_ID	Global partitioned	RANGE (CREW_SHFT_ID)	Plan End Date/Time (PLAN_END_DTTM) for non template shifts (SHFT_USAGE_FLG <> 'M1TM'). Maximum date/time value ('31-DEC-9999') for templates (SHFT_USAGE_FLG = 'M1TM'). Refer to the upgrade script under this table.
		MT100S1	RESRC_ID,PLAN_START_DTTM, PLAN_END_DTTM	Global		
		MT100S2	MDT_ID	Global		
		MT100S3	USER_ID	Global		
		MT100S4	LOGON_MOBILE_PHONE	Global		
		MT100S5	CREW_SHFT_ID, PLAN_START_DTTM,PLAN_END_DTTM,BO_STATUS_CD	Global		
		MT100S6	SHFT_TMPL_ID, PLAN_START_DTTM	Global		
		MT100S7	PLAN_END_DTTM	Global		
		MT100S8	SUBSCRIPTION_ID	Global		
		MT100S9	SHFT_WKTMPL_ID	Global		
		ILM_MT100S10	ILM_DT,ILM_SW_ARCH,CREW_SHFT_ID	Local		
M1_CREW_SHFT_CHAR	Reference partitioning	MT208P0	CREW_SHFT_ID, CHAR_TYPE_CD, SEQ_NUM	Global partitioned	RANGE(CREW_SHFT_ID)	

		MT208S1	SRCH_CHAR_VAL	Global	
M1_CREW_SHFT_DEPOT	Reference partitioning	M1T829P0	CREW_SHFT_ID, M1_DEPOT_CD	Global partitioned	RANGE(CREW_SHFT_ID)
M1_CREW_SHFT_LOC	Reference partitioning	MT201P0	CREW_SHFT_ID, SEQNO	Global partitioned	RANGE(CREW_SHFT_ID)
		MT201S1	LOCATION_CD	Global	
M1_CREW_SHFT_LOG	Reference partitioning	MT204P0	CREW_SHFT_ID, SEQNO	Global partitioned	RANGE(CREW_SHFT_ID)
M1_CREW_SHFT_LOG_PARM	Reference partitioning	MT205P0	CREW_SHFT_ID, SEQNO,PARM_SEQ	Global partitioned	RANGE(CREW_SHFT_ID)
M1_CREW_SHFT_MLOG	Reference partitioning	MT206P0	CREW_SHFT_ID, SEQNO	Global partitioned	RANGE(CREW_SHFT_ID)
M1_CREW_SHFT_MLOG_PARM	Reference partitioning	MT207P0	CREW_SHFT_ID, SEQNO,PARM_SEQ	Global partitioned	RANGE(CREW_SHFT_ID)
M1_CREW_SHFT_RESRC	Reference partitioning	MT200P0	CREW_SHFT_ID, SEQNO	Global partitioned	RANGE(CREW_SHFT_ID)
		MT200S1	RESRC_ID	Global	
M1_CREW_SHFT_SVC_AREA	Reference partitioning	MT202P0	CREW_SHFT_ID, SEQNO	Global partitioned	RANGE(CREW_SHFT_ID)
		MT202S2	CREW_SHFT_ID, SVC_AREA_CD,SEQNO	Global	
M1_CREW_SHFT_SVC_CLS	Reference partitioning	MT203P0	CREW_SHFT_ID, SEQNO	Global partitioned	RANGE(CREW_SHFT_ID)
		MT203S1	SVC_CLS_CD,CREW_SHFT_ID,SEQNO,SVC_CLS_USG_FLG	Global	
M1_CREW_SHFT_SCHED	Reference partitioning	MT209P0	CREW_SHFT_ID	Global partitioned	RANGE(CREW_SHFT_ID)
		MT209S1	SHFT_SCH_STATUS_FLG, CREW_SHFT_ID, COMPLETED_DTTM, VERSION	Global	
		MT209S2	CREW_SHFT_ID, VERSION	Global	
		M1_CREW_SHFT_SCHED_ON_SIDX	M1_LOGON_GEOM	Global	
		M1_CREW_SHFT_SCHED_CAL_SIDX	M1_CAL_GEOM	Global	
		M1_CREW_SHFT_SCHED_OFF_SIDX	M1_LOGOFF_GEOM	Global	

M1_CREW_SHF T_SCHED_ACT UAL_SIDX	M1_ACTUAL_G EOM	Global	

Upgrade Script for Shift (M1_CREW_SHIFT)

```
update M1_CREW_SHFT set ILM_DT=PLAN_END_DTTM where SHFT_USAGE_FLG <>
'M1TM';
update M1_CREW_SHFT set ILM_DT='31-DEC-9999' where
SHFT_USAGE_FLG='M1TM';
commit;
update M1_CREW_SHFT set ILM_ARCH_SW='N';
commit;
```

Task

This table describes the Task maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
M1_TASK (Parent)	RANGE (ILM_DT, TASK_ID)	MT541P0	TASK_ID	Global partitioned	RANGE(TASK_I D)	Creation date/ time(CRE_DTTM) Refer to the upgrade script under this table.
		M1T541S1	M1_FINAL_DTT M	Global		
		MT541S10	RESRC_ID	Global		
		MT541S11	M1_HOST_JOB_ EXTERNAL_ID	Global		
		MT541S12	M1_JOB_ID,M1_ SCHEDULING_ ORDER	Global		
		MT541S13	BO_STATUS_CD, BUS_OBJ_CD,M1 _EAR_SCHD_DT TM,M1_DUE_BY _DTTM,TASK_I D	Global		
		MT541S14	TASK_ID,M1_JO B_ID	Global		
		MT541S15	M1_COMPLEX_ ACTIVITY_ ID	Global		
		MT541S16	M1_REF_TASK_I D	Global		
		MT541S17	M1_LOCKED_T O_DEPOT_TAS K_ID	Global		
		MT541S1	HOST_EXTERN AL_ID	Global		

		MT541S2	PARENT_TASK_ID	Global	
		MT541S3	TASK_ALT_ID	Global	
		MT541S4	PARENT_TASK_ALT_ID	Global	
		MT541S5	CUST_NAME	Global	
		MT541S6	MAN_ALLOC_TO_SHIFT,PARENT_TASK_ID,TASK_TYPE_CD,BO_STATUS_CD,TASK_ID	Global	
		MT541S7	TASK_ID,SVC_AREA_CD,TASK_TYPE_CD,MAN_ALLOC_TO_SHIFT,BO_STATUS_CD,BUS_OBJ_CD	Global	
		MT541S8	BO_STATUS_CD,BUS_OBJ_CD,TASK_ID	Global	
		MT541S9	POU_ID,RESRC_ID	Global	
		ILM_MT541S18	ILM_DT,ILM_ARCH_SW,TASK_ID	Local	
M1_TASK_ADDR	Reference partitioning	MT542P0	TASK_ID,ADDRESS_USAGE_FLG	Global partitioned	RANGE(TASK_ID)
		M1_TASK_ADDR_SIDX	M1_GEOM	Global	
		MT542S1	ADDRESS1	Global	
		MT542S2	CITY	Global	
		MT542S3	COUNTY	Global	
		MT542S4	STATE	Global	
		MT542S5	POSTAL	Global	
		MT542S6	TASK_ID,ADDRESS_USAGE_FLG,GEO_LAT,GEO_LONG	Global	
M1_TASK_CAPABILITY	Reference partitioning	M1T982P0	TASK_ID,CAP_TYPE_CD	Global partitioned	RANGE(TASK_ID)
M1_TASK_CHAR	Reference partitioning	MT569P0	TASK_ID,CHAR_TYPE_CD,SEQ_NUM	Global partitioned	RANGE(TASK_ID)
		MT569S1	SRCH_CHAR_VAL	Global	
M1_TASK_DEPENDENCY	Reference partitioning	M1T835P0	TASK_ID,PARENT_TASK_ID,M1_DEPENDENCY_TYPE_FLG	Global partitioned	RANGE(TASK_ID)

		M1T835S0	PARENT_TASK_ID	Global	
		M1T835S1	M1_HOST_PARENT_EXT_ID	Global	
		M1T835S2	M1_HOST_CHAIN_EXT_ID	Global	
M1_TASK_DEPOT	Reference partitioning	M1T828P0	TASK_ID,M1_DEPOT_CD	Global partitioned	RANGE(TASK_ID)
M1_TASK_LOG	Reference partitioning	MT544P0	TASK_ID,SEQNO	Global partitioned	RANGE(TASK_ID)
		MT544S1	CHAR_TYPE_CD,CHAR_VAL_FK1	Global	
		MT544S2	CHAR_TYPE_CD,CHAR_VAL	Global	
		MT544S3	BO_STATUS_CD,LOG_DTTM	Global	
M1_TASK_LOG_PARM	Reference partitioning	MT545P0	TASK_ID,SEQNO,PARM_SEQ	Global partitioned	RANGE(TASK_ID)
M1_TASK_MOB_LOG	Reference partitioning	MT546P0	TASK_ID,SEQNO	Global partitioned	RANGE(TASK_ID)
		MT546S1	CHAR_TYPE_CD,CHAR_VAL_FK1	Global	
		MT546S2	CHAR_TYPE_CD,CHAR_VAL	Global	
M1_TASK_MOB_LOG_PARM	Reference partitioning	MT568P0	TASK_ID,SEQNO,PARM_SEQ	Global partitioned	RANGE(TASK_ID)
M1_TASK_RESRC_RESTRICTION	Reference partitioning	M1T977P0	TASK_ID,SEQNO	Global partitioned	RANGE(TASK_ID)
M1_TASK_TM_WIND	Reference partitioning	MT543P0	TASK_ID,SEQNO	Global partitioned	RANGE(TASK_ID)
		MT543S1	TASK_ID,TIME_WINDOW_USAGE_FLG,TSK_TM_WIND_END_DTTM,TSK_TM_WIND_START_DTTM,SEQNO	Global	
		MT543S2	TSK_TM_WIND_START_DTTM,TIME_WINDOW_USAGE_FLG,TSK_TM_WIND_END_DTTM,TASK_ID,SEQNO	Global	
		MT543S3	TSK_TM_WIND_END_DTTM	Global	
M1_TASK_SCHEDULE	Reference partitioning	M1T570P0	TASK_ID	Global partitioned	RANGE(TASK_ID)

	MT570S1	SCHDL_SHIFT_ID, M1_SCHED_TASK_CLS_FLG, TASK_ID, VERSION	Global	
	MT570S2	SCHEDULER_STATUS_FLG, TASK_ID, VERSION	Global	
	MT570S3	TASK_ID, SCHDL_SHIFT_ID, VERSION	Global	
	M1T837S2	M1_READY_DTTM, M1_SCHED_TASK_CLS_FLG, SCHEDULER_STATUS_FLG, TASK_ID	Global	
	M1T837S3	ACTUAL_COM_DTTM	Global	
	M1T570S6	SCHEDULER_STATUS_FLG	Global	
	M1_TASK_SCHEDULE_SIDX	M1_GEOM	Global	

Upgrade Script for Task (M1_TASK)

```
update M1_TASK set ILM_DT=CRE_DTTM;
commit;
update M1_TASK set ILM_ARCH_SW='N';
commit;
```

Period of Unavailability

This table describes the Period of Unavailability maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
M1_POU (Parent)	RANGE (ILM_DT, POU_ID)	M1C084P0	POU_ID	Global partitioned	RANGE(POU_ID)	POU End Date/Time (POU_END_DTTM) for non template POUs (POU_USAGE_FLG<>'M1TM'). Maximum date/time value ('31-DEC-9999') for templates (POU_USAGE_FLG='M1TM'). Refer to the upgrade script under this table.

Table Name	Table Partitioning Type	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
		MC084S1	POU_START_DT TM	Global		
		MC084S2	LOCATION_CD	Global		
		ILM_MC084S3	ILM_DT,ILM_AR CH_SW,POU_ID	Local		
M1_POU_CHAR	Reference partitioning	MC089P0	POU_ID,CHAR_ TYPE_CD,SEQ_ NUM	Global partitioned	RANGE(POU_ID)	
		MC089S1	SRCH_CHAR_VA L	Global		
M1_POU_LOG	Reference partitioning	MC087P0	POU_ID,SEQNO	Global partitioned	RANGE(POU_ID)	
		MC087S1	CHAR_TYPE_C D,CHAR_VAL_F K1	Global		
		MC087S2	CHAR_TYPE_C D,CHAR_VAL	Global		
M1_POU_LOG_ PARM	Reference partitioning	MC088P0	POU_ID,SEQNO, PARM_SEQ	Global partitioned	RANGE(POU_ID)	
M1_POU_RESRC	Reference partitioning	MC085P0	POU_ID,RESRC_ ID	Global partitioned	RANGE(POU_ID)	
		MC085S1	RESRC_ID	Global		
M1_POU_SVC_A REA	Reference partitioning	MC086P0	POU_ID,SVC_AR EA_CD	Global partitioned	RANGE(POU_ID)	

Upgrade Script for Period of Unavailability (M1_POU)

```
update M1_POU set ILM_DT=POU_END_DTTM where POU_USAGE_FLG<>'M1TM';
update M1_POU set ILM_DT='31-DEC-9999' where POU_USAGE_FLG ='M1TM';
commit;

update M1_POU set ILM_ARCH_SW='N';
commit;
```

Remote Message

This table describes the Remote Message maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
M1_REMOTE_M SG (Parent)	RANGE (ILM_DT, REMOTE_MSG_ ID)	MT410P0	REMOTE_MSG_ ID	Global partitioned	RANGE(REMOT E_MSG_ID)	Creation date/ time(CRE_DTTM) Refer to the upgrade script under this table.
		MT410S1	STATUS_UPD_D TTM	Global		
		MT410S2	MDT_ID	Global		

		MT410S4	MAINT_OBJ_CD	Global	
		MT410S5	PK_VALUE1	Global	
		MT410S6	DEVICE_MSG_ID	Global	
		MT410S7	MDT_ID,MSG_CLASS_FLG,DELIVERY_STATE_FLG	Global	
		ILM_MT410S8	ILM_DT,ILM_ARCH_SW,REMOTE_MSG_ID	Local	
M1_REMOTE_MSG_CHAR	Reference partitioning	MT413P0	REMOTE_MSG_ID,CHAR_TYPE_CD,SEQ_NUM	Global partitioned	RANGE(REMOTE_MSG_ID)
		MT413S1	SRCH_CHAR_VAL	Global	
M1_REMOTE_MSG_CTXT	Reference partitioning	M1T994P0	REMOTE_MSG_ID,CTXT_NAME	Global partitioned	RANGE(REMOTE_MSG_ID)
		M1T994S1	CTXT_NAME,CTXT_VALUE	Global	
M1_REMOTE_MSG_LOG	Reference partitioning	MT411P0	REMOTE_MSG_ID,SEQNO	Global partitioned	RANGE(REMOTE_MSG_ID)
		MT411S1	CHAR_TYPE_CD,CHAR_VAL_FK1	Global	
		MT411S2	CHAR_TYPE_CD,CHAR_VAL	Global	
M1_REMOTE_MSG_LOG_PARM	Reference partitioning	MT412P0	REMOTE_MSG_ID,SEQNO,PARM_SEQ	Global partitioned	RANGE(REMOTE_MSG_ID)

Upgrade Script for Remote Message (M1_REMOTE_MSG)

```
update M1_REMOTE_MSG set ILM_DT=CRE_DTTM;
commit;
update M1_REMOTE_MSG set ILM_ARCH_SW='N';
commit;
```

Alert

This table describes the Alert maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
M1_ALERT (Parent)	RANGE (ILM_DT, ALERT_ID)	MT077P0	ALERT_ID	Global partitioned	RANGE(ALERT_ID)	Creation date/time(CRE_DTTM) Refer to the upgrade script under this table.

Table Name		Index Name	Index Columns	Index Type Global or Local	Index Partitioning Key
		MT077S1	EFF_DTTM	Global	
		MT077S2	BO_STATUS_CD	Global	
		MT077S3	USER_ID	Global	
		MT077S4	CRE_DTTM	Global	
		MT077S5	TASK_ID	Global	
		MT077S6	CREW_SHFT_ID	Global	
		MT077S7	RESRC_ID	Global	
		ILM_MT077S8	ILM_DT,ILM_ARCH_SW,ALERT_ID	Local	
M1_ALERT_CHAR	Reference partitioning	MT065P0	ALERT_ID,CHAR_TYPE_CD,SEQ_NUM	Global partitioned	RANGE(ALERT_ID)
		MT065S1	SRCH_CHAR_VAL	Global	
M1_ALERT_LOG	Reference partitioning	MT078P0	ALERT_ID,SEQ NO	Global partitioned	RANGE(ALERT_ID)
		MT078S1	CHAR_TYPE_CD,CHAR_VAL_FK1	Global	
		MT078S2	CHAR_TYPE_CD,CHAR_VAL	Global	
M1_ALERT_LOG_PARM	Reference partitioning	MT064P0	ALERT_ID,SEQ NO,PARM_SEQ	Global partitioned	RANGE(ALERT_ID)

Upgrade Script for Alert (M1_ALERT)

```
update M1_ALERT set ILM_DT=CRE_DTTM;
commit;
update M1_ALERT set ILM_ARCH_SW='N';
commit;
```

Dispatcher Shift

This table describes the Dispatcher Shift maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
M1_DISP_SHIFT (Parent)	RANGE (ILM_DT, DISP_SHIFT_ID)	MT310P0	DISP_SHIFT_ID	Global partitioned	RANGE(DISP_SHIFT_ID)	Creation date/time(CRE_DTTM) Refer to the upgrade script under this table.
		MT310S1	BO_STATUS_CD	Global		

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
		MT310S2	RESRC_ID,LOG OFF_DTTM,LO GON_DTTM,SC P_END_DTTM,S CP_START_DTT M,DISP_SHIFT_I D	Global		
		MT310S3	LOGOFF_DTTM	Global		
		MT310S4	LOGON_DTTM	Global		
		MT310S5	SCP_START_DT TM	Global		
		MT310S6	SCP_END_DTT M	Global		
		MT310S7	BUS_OBJ_CD	Global		
		ILM_MT310S8	ILM_DT,ILM_AR CH_SW,DISP_SH IFT_ID	Local		
M1_DISP_SHIFT _CHAR	Reference partitioning	MT314P0	DISP_SHIFT_ID, CHAR_TYPE_C D,SEQ_NUM	Global partitioned	RANGE(DISP_S HIFT_ID)	
		MT314S1	SRCH_CHAR_VA L	Global		
M1_DISP_SHIFT _LOG	Reference partitioning	MT312P0	DISP_SHIFT_ID, SEQNO	Global partitioned	RANGE(DISP_S HIFT_ID)	
		MT312S1	CHAR_TYPE_C D,CHAR_VAL_F K1	Global		
		MT312S2	CHAR_TYPE_C D,CHAR_VAL	Global		
M1_DISP_SHIFT _LOG_PARM	Reference partitioning	MT313P0	DISP_SHIFT_ID, SEQNO,PARM_S EQ	Global partitioned	RANGE(DISP_S HIFT_ID)	
M1_DISP_SHIFT _MON_AREA	Reference partitioning	MT311P0	DISP_SHIFT_ID, DISP_AREA_CD	Global		
		MT311S1	DISP_AREA_CD, DISP_SHIFT_ID	Global		

Upgrade Script for Dispatcher Shift (M1_DISP_SHIFT)

```
update M1_DISP_SHIFT set ILM_DT=CRE_DTTM;
commit;
update M1_DISP_SHIFT set ILM_ARCH_SW='N';
commit;
```

Mail

This table describes the Mail maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load

M1_MAIL (Parent)	RANGE (ILM_DT, M1_MAIL_ID)	M1T983P0	M1_MAIL_ID	Global partitioned	RANGE(M1_MAIL_ID)	Creation Date/ time (CRE_DTTM) for non template mail records (M1_MAIL_MSG _CLS_FLG<>'M1 TE'). Maximum date/ time value ('31-DEC-9999') for templates (M1_MAIL_MSG _CLS_FLG='M1T E'). Refer to the upgrade script under this table.
		M1T983S0	BO_STATUS_CD, BUS_OBJ_CD,M1 _MAIL_ID	Global		
		M1T983S1	USER_ID,M1_FO LDER_FLG,CRE _DTTM,M1_MLR D_STATUS_FLG	Global		
		M1T983S2	CREW_SHFT_ID, M1_FOLDER_FL G,CRE_DTTM,M 1_MLRD_STATU S_FLG	Global		
		M1T983S3	M1_SENT_BY_M AIL_ID	Global		
		ILM_M1T983S4	ILM_DT,ILM_AR CH_SW,M1_MAI L_ID	Local		
M1_MAIL_CHAR	Reference partitioning	M1T987P0	M1_MAIL_ID,CH AR_TYPE_CD,SE Q_NUM	Global partitioned	RANGE(M1_MAI L_ID)	
M1_MAIL_LOG	Reference partitioning	M1T985P0	M1_MAIL_ID,SE QNO	Global partitioned	RANGE(M1_MAI L_ID)	
M1_MAIL_LOG_ PARM	Reference partitioning	M1T986P0	M1_MAIL_ID,SE QNO,PARM_SEQ	Global partitioned	RANGE(M1_MAI L_ID)	
M1_MAIL_SEND _TO	Reference partitioning	M1T984P0	M1_MAIL_ID,SE QNO	Global partitioned	RANGE(M1_MAI L_ID)	
		M1T984S0	USER_ID	Global		
		M1T984S1	CREW_SHFT_ID	Global		
		M1T984S2	M1_MAILING_L IST_CD	Global		
		M1T984S3	M1_PRDF_SNDT O_FLG	Global		

Upgrade Script for Mail (M1_MAIL)

update M1_MAIL set ILM_DT=CRE_DTTM where M1_MAIL_MSG_CLS_FLG<>'M1TE';

update M1_POU set ILM_DT='31-DEC-9999' where M1_MAIL_MSG_CLS_FLG ='M1TE';

commit;

update M1_MAIL set ILM_ARCH_SW='N';

commit;

Procedure

This table describes the Procedure maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
M1_PROCEDUR E (Parent)	RANGE (ILM_DT, PROCEDURE_I D)	M1T839P0	PROCEDURE_I D	Global partitioned	RANGE(PROCE DURE_ID)	Creation date/ time(CRE_DTTM) Refer to the upgrade script under this table.
		M1T839S1	PROCEDURE_T YPE_CD	Global		
		M1T839S2	CRE_DTTM	Global		
		ILM_M1T839S3	ILM_DT,ILM_AR CH_SW,PROCED URE_ID	Local		
M1_PROCEDUR E_CHAR	Reference partitioning	M1T844P0	PROCEDURE_I D,CHAR_TYPE_ CD,SEQ_NUM	Global partitioned	RANGE(PROCE DURE_ID)	
M1_PROCEDUR E_LOG	Reference partitioning	M1T840P0	PROCEDURE_I D,SEQNO	Global partitioned	RANGE(PROCE DURE_ID)	
M1_PROCEDUR E_LOG_PARM	Reference partitioning	M1T841P0	PROCEDURE_I D,SEQNO,PARM _SEQ	Global partitioned	RANGE(PROCE DURE_ID)	
M1_PROCEDUR E_MOB_LOG	Reference partitioning	M1T842P0	PROCEDURE_I D,SEQNO	Global partitioned	RANGE(PROCE DURE_ID)	
M1_PROCEDUR E_MOB_LOG_P ARM	Reference partitioning	M1T843P0	PROCEDURE_I D,SEQNO,PARM _SEQ	Global partitioned	RANGE(PROCE DURE_ID)	

Upgrade Script for Procedure (M1_PROCEDURE)

update M1_PROCEDURE set ILM_DT=CRE_DTTM;

commit;

update M1_PROCEDURE set ILM_ARCH_SW='N';

commit;

Timesheet

This table describes the Timesheet maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
M1_TIMESHEET (Parent)	RANGE (ILM_DT, TIMESHEET_ID)	M1T850P0	TIMESHEET_ID	Global partitioned	RANGE(TIMESHEET_ID)	Creation date/time(CRE_DTTM) Refer to the upgrade script under this table.
		M1T850S1	CREW_SHFT_ID	Global		
		M1T850S2	RESRC_ID	Global		
		M1T850S3	BO_STATUS_CD	Global		
		M1T850S4	CRE_DTTM	Global		
		ILM_M1T850S5	ILM_DT,ILM_ARCH_SW,TIMESHEET_ID	Local		
M1_TIMESHEET _CHAR	Reference partitioning	M1T857P0	TIMESHEET_ID, CHAR_TYPE_CD,SEQ_NUM	Global partitioned	RANGE(TIMESHEET_ID)	
		M1T857S1	SRCH_CHAR_VAL	Global		
M1_TIMESHEET _LINE	Reference partitioning	M1T852P0	TIMESHEET_ID, TIMESHEET_TM_TYPE_CD	Global partitioned	RANGE(TIMESHEET_ID)	
M1_TIMESHEET _LOG	Reference partitioning	M1T853P0	TIMESHEET_ID, SEQNO	Global partitioned	RANGE(TIMESHEET_ID)	
		M1T853S1	CHAR_TYPE_CD,CHAR_VAL_FK1	Global		
		M1T853S2	CHAR_TYPE_CD,CHAR_VAL	Global		
M1_TIMESHEET _LOG_PARM	Reference partitioning	M1T854P0	TIMESHEET_ID, SEQNO,PARM_SEQ	Global partitioned	RANGE(TIMESHEET_ID)	
M1_TIMESHEET _MLOG	Reference partitioning	M1T855P0	TIMESHEET_ID, SEQNO	Global partitioned	RANGE(TIMESHEET_ID)	
M1_TIMESHEET _MLOG_PARM	Reference partitioning	M1T856P0	TIMESHEET_ID, SEQNO,PARM_SEQ	Global partitioned	RANGE(TIMESHEET_ID)	

Upgrade Script for Timesheet (M1_TIMESHEET)

update M1_TIMESHEET set ILM_DT=CRE_DTTM;

commit;

update M1_TIMESHEET set ILM_ARCH_SW='N';

commit;

GPS Data

This table describes the GPS Data maintenance object.

Table Name	Table Partitioning Type (Partitioning, Sub-Partitioning Key)	Index Name	Index Columns	Index Type Global or Local	Index Partitioning Sub-Partitioning Key	ILM_DT Initial Load
M1_GPS_DATA	RANGE (ILM_DT, GPS_DATA_ID)	MT315P0	GPS_DATA_ID	Global partitioned	RANGE(GPS_DATA_ID)	
		M1_GPS_DATA_SIDX	M1_GEOM	Global		
		MT315S1	RESRC_ID,CAP_DTTM	Global		
		MT315S2	CAP_DTTM	Global		
		MT315S3	EXT_SRC_TYPE_FLG,EXT_SRC_ID	Global		

On-going Maintenance Phase

As mentioned in the overview, for each maintenance object that supports ILM, there are batch processes that periodically review "old" records and mark the ILM Archive switch to "Y".

Once the ILM fields are populated with their relevant values, the ILM facilities are then used within the database to implement storage savings, as per the following:

- Use ILM Assistant to define the data groups to be used for the individual objects. Assign those data groups to partitions and storage devices to implement the storage savings. Remember to assign transportable tablespaces for the archive/dormant data stage to allow for safe removal of the data.

- Use ILM assistant to generate the necessary commands to implement the data changes manually or use Automatic Storage Management (ASM) to automate the data storage policies.

- Optionally, use Automatic Data Optimization to provide further optimizations.

The following figure illustrates the steps necessary for this phase:
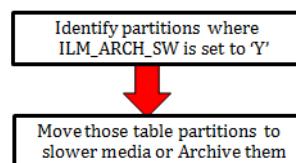
For more information about ILM Assistant and ILM refer to the following:

- ILM Assistant Users Guide available at:

  http://download.oracle.com/otn/other/ilm/ilma-users-guide.html

- Oracle Database VLDB and Partitioning Guide (11.2) available at:

    http://docs.oracle.com/cd/E11882_01/server.112/e25523/
    part_lifecycle.htm#CACECAFB

- Oracle Database VLDB and Partitioning Guide (12.1) available at:

    https://docs.oracle.com/database/121/VLDBG/title.htm

ILM Assistant

The ILM Assistant in the current 11g database implementation can provide the following:

- Setup ILM Lifecycle definition - Here you can define different lifecycle definitions for different MOs and say that after what period of time the data is ready to be moved to a slower disk.

- Setup ILM Lifecycle tables - Here you define the tables you want to manage and assign it to a Lifecycle definition defined above. You can setup policies that when data is moved from one partition to another it should be automatically compressed.

- Lifecycle Management - There is a tab called Lifecycle Management where the system admin will be alerted for when the partitions are eligible for archiving.

ILM Assistant can then be used with the ILM to make sure the records that have ILM_ARCH_SW = 'Y' can be moved to slower and slower disks and possibly get purged.
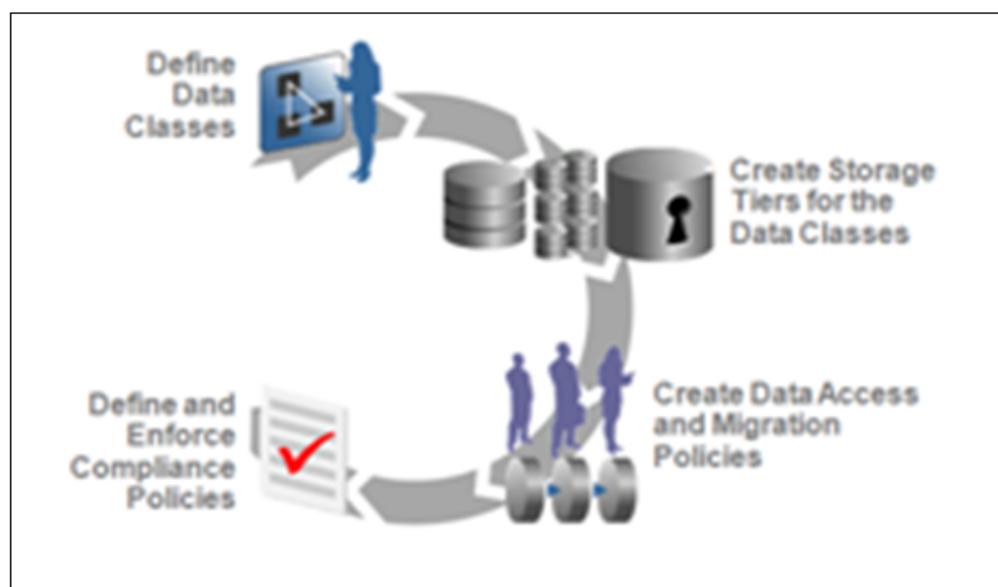
Note: For further guidelines on ILM Assistant, refer to Implementing Information Lifecycle Management Using the ILM Assistant available at:

http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/prod/storage/ilm/
ilm.htm?cid=4196&ssid=115606280996764

Appendix A

Upgrades to the Oracle Utilities Application Framework Database

This section describes the database upgrade process for the Oracle Utilities Application Framework database since the last release. It highlights changes made to the administrative tables and how those changes should be applied to the data in order for your current database to work with the Oracle Utilities Application Framework application, and to preserve the business logic implemented in the previous version of the application. The changes that do not require data upgrade are not described in this document. The tasks that need to be performed after running the upgrade scripts are included.

> **Note**: Upgrade scripts do not automatically enable the newly added functionality by default. Please refer to the release notes for more information.

The section provides information on upgrading the Oracle Utilities Application Framework Database including:

- Upgrading from Oracle Utilities Application Framework v4.3.0.1.0 to v4.3.0.4.0
- Upgrading from Oracle Utilities Application Framework v4.3.0.2.0 to v4.3.0.4.0
- Upgrading from Oracle Utilities Application Framework v4.3.0.3.0 to v4.3.0.4.0

Upgrading from Oracle Utilities Application Framework v4.3.0.1.0 to v4.3.0.4.0

New Tables

Table	Type of Table
F1_BUS_FLG	Business Flag
F1_BUS_FLG_CHAR	Business Flag Characteristic
F1_BUS_FLG_K	Business Flag Key Table
F1_BUS_FLG_LOG	Business Flag Log
F1_BUS_FLG_LOG_PARM	Business Flag Log Parameter

Table	Type of Table
F1_BUS_FLG_REL	Business Flag Relationship
F1_BUS_FLG_REL_OBJ	Business Flag Related Object
F1_BUS_FLG_TYPE	Business Flag Type
F1_BUS_FLG_TYPE_ALG	Business Flag Type Algorithm
F1_BUS_FLG_TYPE_BUS_PROC	Business Flag Type / Business Process
F1_BUS_FLG_TYPE_CHAR	Business Flag Type Characteristic
F1_BUS_FLG_TYPE_L	Business Flag Type Language
F1_ETL_MP_CTRL	ETL Mapping Control

New Views

None

Dropped Tables

None

Unsupported Tables

None

Added Columns

Table	Column	Required
CI_BATCH_THD	LOG_FILE_NAME	
CI_MD_TBL	CHAR_ENTITY_FLG	N
F1_EXTSYS_OUTMSG_PROF	JSON_CONVRSN_METH_FLG	N
F1_EXTSYS_OUTMSG_PROF	REQ_SCHEMA_NAME	N
F1_EXTSYS_OUTMSG_PROF	RES_SCHEMA_NAME	N
F1_MIGR_OBJ_ALG	ALG_PROC_TYPE_FLG	N

Dropped Columns

Table	Column
CI_MD_TBL	TBL_USAGE_FLG
CI_MD_TBL_FLD	FLD_USAGE_FLG

Unsupported Table Columns

None

Column Format Change

None

Upgrading from Oracle Utilities Application Framework v4.3.0.2.0 to v4.3.0.4.0

New Tables

Table	Type of Table
F1_LGCY_OBJ	Legacy Object
F1_PERF_TGT	Performance Target
F1_PERF_TGT_CHAR	Performance Target Characteristic
F1_PERF_TGT_L	Performance Target Language
F1_PERF_TGT_LOG	Performance Target Log
F1_PERF_TGT_LOG_PARM	Performance Target Log Parameter
F1_PERF_TGT_REL_OBJ	Performance Target Related Object
F1_PERF_TGT_TYPE	Performance Target Type
F1_PERF_TGT_TYPE_CHAR	Performance Target Type Characteristic
F1_PERF_TGT_TYPE_L	Performance Target Type Language
F1_STATS	Statistics Control
F1_STATS_CHAR	Statistics Control Characteristic
F1_STATS_L	Statistics Control Language
F1_STATS_LOG	Statistics Control Log
F1_STATS_LOG_PARM	Statistics Control Log Parameter
F1_STATS_REL_OBJ	Statistics Control Related Object
F1_STATS_SNPSHT	Statistics Snapshot
F1_STATS_SNPSHT_CHAR	Statistics Snapshot Characteristic
F1_STATS_SNPSHT_LOG	Statistics Snapshot Log
F1_STATS_SNPSHT_LOG_PARM	Statistics Snapshot Log Parameter
F1_STATS_SNPSHT_REL_OBJ	Statistics Snapshot Related Object
F1_SVC_CATALOG	Web Service Catalog

New Views

None

Dropped Tables

None

Unsupported Tables

None

Added Columns

Table	Column	Required
F1_EXTSYS_OUTMSG_PROF	NAMESPACE_FLG	N
F1_EXTSYS_OUTMSG_PROF	WSDL_FILE_NAME	N

Dropped Columns

None

Unsupported Table Columns

None

Column Format Change

None

Primary Key Change

None

Upgrading from Oracle Utilities Application Framework v4.3.0.3.0 to v4.3.0.4.0

New Tables

Table	Type of Table
F1_MIGR_REQ_INCL_REQ	Migration request Grouping

New Views

None

Dropped Tables

None

Unsupported Tables

None

Added Columns

Table	Column	Required
CI_BATCH_CTRL	APP_SVC_ID	Y
CI_XAI_RCVR_CTX	SEQNO	Y
CI_XAI_SNDR_CTX	SEQNO	Y
F1_IWS_SVC_ANN	SEQ_NUM	Y
F1_MIGR_REQ	MIGR_REQ_CAT_XFLG	N
F1_MIGR_REQ	MIGR_REQ_CLASS_FLG	Y
F1_MIGR_REQ_INSTR_ENTITY	COMMENT_LONG	N
F1_MIGR_REQ_INSTR_ENTITY	EXT_REFERENCE_ID	N

Dropped Columns

Table	Column
CI_XAI_RCVR_CTX	CTXT_VAL

Unsupported Table Columns
None

Column Format Change

Table Name	Column Name	From	To
F1_EXT_LOOKUP_VAL_CHAR	F1_EXT_LOOKUP_VALUE	VARCHAR2 (30)	VARCHAR2 (254)

Primary Key Change

Table	Primary Key Columns
CI_XAI_RCVR_CTX	XAI_RCVR_ID, SEQNO
CI_XA_SNDR_CTX	XAI_SENDER_ID, SEQNO

Appendix B

Oracle Utilities Mobile Workforce Management System Table Guide

This chapter lists the system tables owned by Oracle Utilities Mobile Workforce Management and explains the data standards of the system tables. The data standards are required for Oracle Utilities Mobile Workforce Management installation, development within Oracle Utilities Mobile Workforce Management, configuration of Oracle Utilities products, and customization of the Oracle Utilities products. Adhering to the data standards is a prerequisite for a seamless upgrade to the next release of the product. For the general discussion of System Tables, refer to the System Table Guide for the Oracle Utilities Application Framework.

Development and Implementation System Tables

Parameter Definition

Properties	Description
Tables	M1_PARM_DEF
	M1_PARM_DEF_L
	M1_DEPLOYMENT_PART
	M1_DEPLOYMENT_PART_L
	M1_DEPLOYMENT_ITEM
Standard Data Fields	Parameter Definition (PARM_DEF_CD)
Customer Modification	None

Appendix C

Oracle Utilities Application Framework System Table Guide

This section lists the system tables owned by the Oracle Utilities Application Framework V4.3.0.4.0 and explains the data standards of the system tables. The data standards are required for the installation of Oracle Utilities Application Framework, development within the Oracle Utilities Application Framework, and the configuration and customization of Oracle Utilities products. Adhering to the data standards is a prerequisite for seamless upgrade to future releases.

This section includes:

- About the Application Framework System Tables

- System Table Standards

- Guidelines for System Table Updates

- System Table List

About the Application Framework System Tables

System tables are a subset of the tables that must be populated at the time the product is installed. They include metadata and configuration tables. The data stored in the system tables are the information that Oracle Utilities Application Framework product operations are based on.

As the product adds more functionality, the list of system tables can grow. The complete list of the system tables can be found in the System Table List section.

System Table Standards

System table standards must be observed for the following reasons:

- The product installation and upgrade process and customer modification data extract processes depend on the data prefix and owner flag values to determine the system data owned by each product.

- The standards ensure that there will be no data conflict in the product being developed and the future Oracle Utilities Application Framework release. They also ensure that there will be no data conflict between customer modifications and future Oracle Utilities product releases.

- The data prefix is used to prevent test data from being released to production.

  Developer's Note: All test data added to the system data tables must be prefixed by ZZ (all upper case) in order for the installation and upgrade utility to recognize them as test data.

Guidelines for System Table Updates

This section describes guidelines regarding the updating of the system table properties.

Business Configuration Tables

The majority of data in the tables in this group belongs to the customer. But these tables are shipped with some initial data in order for the customer to login to the system and begin configuring the product. Unless specified otherwise, the initial data is maintained by Oracle Utilities Application Framework and subject to subsequent upgrade.

Application Security and User Profile

These tables define the access rights of a User Group to Application Services and Application Users.

Properties	Description
Tables	SC_ACCESS_CNTL, SC USER, SC_USR_GRP_PROF, SC_USR_GRP_USR, SC_USER_GROUP, SC_USER_GROUP_L
Initial Data	User Group ALL_SERVICES and default system user SYSUSER. Upon installation the system default User Group ALL_SERVICES is given unrestricted accesses to all services defined in Oracle Utilities Application Framework.

> **Developer's Note:** When a new service is added to the system, all actions defined for the service must be made available to the User Group ALL_ SERVICES.

Currency Code

The ISO 4217 three-letter codes are taken as the standard code for the representation of each currency.

Properties	Description
Tables	CI_CURRENCY_CD, CI_CURRENCY_CD_L
Initial Data	United States Dollar (USD)

Display Profile

The Display Profile Code is referenced in the User (SC_USER) table.

Properties	Description
Tables	CI_DISP_PROF, CI_DISP_PROF_L

Properties	Description
Initial Data	North America (NORTHAM) and Europe (EURO) and HIJRI Format (HIJRI)
	Configuration Note: In order to use HIJRI Format display profile, additional configuration is needed to define the mappings between Hijri dates and Gregorian dates.
	Refer to the Display Profile documentation for more information.

Configuration Note: In order to use HIJRI Format display profile, additional configuration is needed to define the mappings between Hijri dates and Gregorian dates.
Refer to the Display Profile documentation for more information.

Installation Options

Installation Option has only one row that is shipped with the initial installation of the Oracle Utilities Application Framework. The updatable columns in these tables are customer data and will not be overridden by the upgrade process unless a special script is written and included in the upgrade process.

Properties	Description
Tables	F1_INSTALLATION, CI_INSTALL_ALG, CI_INSTALL_MSG, CI_INSTALL_MSG_L, CI_INSTALL_PROD
Initial Data	Option 11111

> **Developer's Note:** The system data owner of an environment is defined in the Installation Option. This Owner Flag value is stamped on all system data that is added to this environment. The installation default value is Customer Modification (CM). This value must be changed in the base product development environments.

Language Code

Language Code must be a valid code defined in ISO 639-2 Alpha-3. Adding a new language code to the table without translating all language dependent objects in the system can cause errors when a user chooses the language.

Properties	Description
Tables	CI_LANGUAGE
Initial Data	English (ENG)

To Do Priority and Role

New To Do Types released will be linked to the default To Do Role and set to the product assigned priority value initially. These initial settings can be overridden by the implementation.

Properties	Description
Tables	CI_ROLE(L), CI_TD_VAL_ROLE
Initial Data	F1_DFLT

Development and Implementation System Tables

This section defines the standards for the system tables that contain data for application development. The data in these tables implement business logic and UI functions shared by various products and product extensions in the same database.

Standards

When adding new data, the owner flag value of the environment must prefix certain fields of these tables. For example, when a developer adds a new algorithm type to an <Product Name> environment, C1 should prefix the new Algorithm Type code. The fields that are subject to this rule are listed in Standard Data Fields property.

The data that is already in these tables cannot be modified if the data owner is different than the environment owner. This prevents the developers from accidentally modifying system data that belongs to the Oracle Utilities Application Framework or the base products. However, some fields are exempt from this rule and can be modified by Customer Modification. These fields are listed in the Customer Modification Fields property.

Note that the system supports a system upgrade rule called Override Owner flag. If duplicate data rows (data row with same primary key values) are found at the time of upgrade, the owner flag values will get overridden. The lower level application system data will override the upper level system data. For example, F1 overrides C1, F1&C1 override CM, and so on. This rule will be applied to the following tables: CI_CHAR_ENTITY, CI_MD_MO_ALG, C1_PORTAL_OPT, F1_BUS_OBJ_ALG, F1_BUS_OBJ_STATUS_ALG, CI_MD_MO_OPT, F1_BUS_OBJ_OPT, F1_BUS_OBJ_STATUS_OPT, F1_BUS_OBJ_STATUS, F1_BUS_OBJ_STATUS_L

Algorithm Type

Properties	Description
Tables	CI_ALG_TYPE, CI_ALG_TYPE_L, CI_ALG_TYPE_PRM, CI_ALG_TYPE_PRM_L
Standard Data Fields	Algorithm Type (ALG_TYPE_CD)
Customer Modification	None

Algorithm

Properties	Description
Tables	CI_ALG, CI_ALG_L, CI_ALG_PARM, CI_ALG_VER
Standard Data Fields	Algorithm (ALG_CD)
Customer Modification	None

Application Security

Properties	Description
Tables	SC_APP_SERVICE, SC_APP_SERVICE_L, CI_APP_SVC_ACC
Standard Data Fields	Application Service ID (APP_SVC_ID).
Customer Modification	None

Batch Control

Properties	Description
Tables	CI_BATCH_CTRL, CI_BATCH_CTRL_L, CI_BATCH_CTRL_P, CI_BATCH_CTRL_P_L
Standard Data Fields	Batch Process (BATCH_CD), Program Name (PROGRAM_NAME)
Customer Modification	Next Batch Number (NEXT_BATCH_NBR), Last Update Instance (LAST_UPDATE_INST), Last Update Date time (LAST_UPDATE_DTTM) and the batch process update these columns. Time Interval (TIMER_INTERVAL), Thread Count (BATCH_THREAD_CNT), Maximum Commit Records (MAX_COMMIT_RECS), User (USER_ID), Language (LANGUAGE_CD), Email Address (EMAILID), Start program debug tracing (TRC_PGM_STRT_SW), End Program Debug trace (TRC_PGM_END_SW), SQL debug tracing (TRC_SQL_SW) and Standard debug tracing (TRC_STD_SW) on CI_BATCH_CTRL Table. Batch Parameter Value (BATCH_PARM_VAL) and Security flag (TEXT_SECURITY_FLG) on Batch Control Parameters Table (CI_BATCH_CTRL_P)

Business Object

Properties	Description
Tables	F1_BUS_OBJ, F1_BUS_OBJ_L, F1_BUS_OBJ_ALG, F1_BUS_OBJ_OPT, F1_BUS_OBJ_STATUS, F1_BUS_OBJ_STATUS_L, F1_BUS_OBJ_STATUS_ALG, F1_BUS_OBJ_STATUS_OPT, F1_BUS_OBJ_STATUS_RSN, F1_BUS_OBJ_STATUS_RSN_L, F1_BUS_OBJ_STATUS_RSN_CHAR, F1_BUS_OBJ_TR_RULE, F1_BUS_OBJ_TR_RULE_L
Standard Data Fields	Business Object (BUS_OBJ_CD), Status Reason (BO_STATUS_REASON_CD)
Customer Modification	Batch Control (BATCH_CD), Alert (BO_ALERT_FLG), Sequence (SORT_SEQ5), Status Reason (STATUS_REASON_FLG) fields on Business Object Status Table (F1_BUS_OBJ_STATUS). Instance Control (INSTANCE_CTRL_FLG), Application Service (APP_SVC_ID) on Business Object Table (F1_BUS_OBJ). Status Reason Selection (STATUS_REASON_SELECT_FLG) on Status Reason Table (F1_BUS_OBJ_STATUS_RSN)

Business Service

Properties	Description
Tables	F1_BUS_SVC, F1_BUS_SVC_L
Standard Data Fields	Business Service (BUS_SVC_CD)
Customer Modification	Application Service (APP_SVC_ID)

Characteristics

Properties	Description
Tables	CI_CHAR_TYPE, CI_CHAR_TYPE_L, CI_CHAR_ENTITY, CI_CHAR_VAL, CI_CHAR_VAL_L
Standard Data Fields	Characteristic Type (CHAR_TYPE_CD) Characteristic Value (CHAR_VAL) on CI_CHAR_VAL If the characteristic type is customizable, Customer Modification can insert new characteristic values. CM must prefix when implementers introduce a new characteristic value.
Customer Modification	Adhoc Characteristic Value Validation Rule (ADHOC_VAL_ALG_CD), Allow Search by Characteristic Value (SEARCH_FLG)

Configuration Migration Assistant

Properties	Description
Tables	F1_MIGR_PLAN,F1_MIGR_PLAN_L,F1_MIGR_PLAN_IN STR,F1_MIGR_PLAN_INSTR_L, F1_MIGR_PLAN_INSTR_ALG, F1_MIGR_REQ, F1_MIGR_REQ_L, F1_MIGR_REQ_INSTR, F1_MIGR_REQ_INST R_L, F1_MIGR_REQ_INSTR_ENTITY, F1_MIGR_REQ_INCL_REQ
Standard Data Fields	Migration Plan Code (MIGR_PLAN_CD), Migration Request Code (MIGR_REQ_CD)
Customer Modification	None

Data Area

Properties	Description
Tables	F1_DATA_AREA, F1_DATA_AREA_L
Standard Data Fields	Data Area Code (DATA_AREA_CD)
Customer Modification	None

Display Icon

Properties	Description
Tables	CI_DISP_ICON, CI_DISP_ICON_L
Standard Data Fields	Display Icon Code (DISP_ICON_CD)
Customer Modification	None

Extendable Lookup

Properties	Description
Tables	F1_EXT_LOOKUP_VAL, F1_EXT_LOOKUP_VAL_L, F1_EXT_LOOKUP_VAL_CHAR
Standard Data Fields	Business Object (BUS_OBJ_CD), Extendable Lookup Value (F1_EXT_LOOKUP_VALUE)

Properties	Description
Customer Modification	Business Object Data Area (BO_DATA_AREA) Override Description (DESCR_OVRD) on Extendable Lookup Field Value Language Table (F1_EXT_LOOKUP_VAL_L) **Note:** When the product releases base owned records in Extendable Lookup, if there are additional elements the business object will map the element to the BO_DATA_AREA if the value is allowed to be modified by an implementation.

Foreign Key Reference

Properties	Description
Tables	CI_FK_REF, CI_FK_REF_L
Standard Data Fields	FK reference code (FK_REF_CD)
Customer Modification	Info Program Name (INFO_PRG), Zone (ZONE_CD)

Inbound Web Service

Properties	Description
Tables	F1_IWS_SVC_L, F1_IWS_SVC, F1_IWS_SVC_OPER_L, F1_IWS_SVC_OPER, F1_IWS_ANN_L, F1_IWS_ANN_PARM, F1_IWS_ANN, F1_IWS_ANN_TYPE_L, F1_IWS_ANN_TYPE, F1_IWS_ANN_TYPE_PARM, F1_IWS_ANN_TYPE_PARM_L
Standard Data Fields	Webservice Name (IN_SVC_NAME), Annotation (ANN_CD), Annotation Type (ANN_TYPE_CD)
Customer Modification	Debug (DEBUG_SW), Active (ACTIVE_SW), Trace (TRACE_SW), Request XSL (REQUEST_XSL), Response XSL (RESPONSE_XSL)

Legacy Object

Properties	Description
Tables	F1_LGCY_OBJ
Standard Data Fields	Legacy Object ID (LGCY_OBJ_ID)
Customer Modification	

Lookup

Properties	Description
Tables	CI_LOOKUP_FIELD, CI_LOOKUP_VAL, CI_LOOKUP_VAL_L
Standard Data Fields	Field Name (FIELD_NAME) • A lookup field name must have corresponding field metadata. The name of the lookup field column must be assigned to avoid conflicts among different products. If you follow the standards for database field names, a Customer Modification lookup field name will be automatically Customer Modification prefixed. Field Value (FIELD_VALUE) • If a lookup field is customizable, Customer Modification can insert new lookup values. X or Y must prefix when implementers introduce a new lookup value. • Product development may add lookup values to a Oracle Utilities Application Framework owned lookup field's value. When extended new value is added, the Owner Flag is used to prefix the value. .
Customer Modification	Override Description (DESCR_OVRD) on Lookup Field Value Language Table (CI_LOOKUP_VAL_L)

Map

Properties	Description
Tables	F1_MAP, F1_MAP_L
Standard Data Fields	UI Map (MAP_CD)
Customer Modification	None

Managed Content

Properties	Description
Tables	F1_MANAG_CONTENT, F1_MANAG_CONTENT_L
Standard Data Fields	Managed Content (MANAG_CONTENT_CD)
Customer Modification	None

Messages

Properties	Description
Tables	CI_MSG_CATEGORY, CI_MSG_CATEGORY_L, CI_MSG, CI_MSG_L
Standard Data Fields	Message Category (MESSAGE_CAT_NBR) • Messages are grouped in categories and each category has message numbers between 1 and 99999. A range of message categories is assigned to a product. An implementation may only use categories assigned for customization use. • Implementer Message Categories are 80000 and 90000 • Reserved for Tests - 99999 Message Number (MESSAGE_NBR) for message categories • Message numbers below 1000 are reserved for common messages. Implementers must not use message numbers below 1000. Message Number (MESSAGE_NBR) for Java message categories • Subsystem Standard Messages - 00001 thru 02000 • Reserved - 02001 thru 09999 • Published Messages - 10001 thru 11000 • Package Messages - 10001 thru 90000 • Reserved - 90001 thru 99999 • Each package is allocated 100 message numbers, each starting from 101. • Published Messages are messages that are special-interest messages that implementations need to know about and are therefore published in the user docs. Examples of these include messages that are highly likely to be changed for an implementation, or messages that are embedded into other texts/messages and therefore the message number is never shown • Reserved message number ranges are for future use and therefore must not be used by all products.
Customer Modification	Override Description (DESCRLONG_OVRD), Message Text Override (MESSAGE_TEXT_OVRD)

Meta Data - Table and Field

Properties	Description
Tables	CI_MD_TBL, CI_MD_TBL_FLD, CI_MD_TBL_L, CI_MD_TBL_FLD_L, CI_MD_FLD, CI_MD_FLD_L, F1_DB_OBJECTS_REPO
Standard Data Fields	Table Name (TBL_NAME) • Table names must match with the physical table name or view name in the database. • Field Name (FLD_NAME) Field name must match with the physical column name in the database unless the field is a work field. Field name does not have to follow the prefixing standard unless the field is a work field or customer modification field. • F1_DB_OBJECTS_REPO Table stores information about Indexes, Sequences, Triggers and other database objects excluding Tables and Fields (as they are already stored in the other Metadata tables)
Customer Modification	AuditSwitches(AUDIT_INSERT_SW,AUDIT_UPDATE_SW, AUDIT_DELETE_SW), Override label (OVRD_LABEL) on MD Table Field Table (CI_MD_TBL_FLD). Audit Program Name (AUDIT_PGM_NAME), Audit Table Name (AUDIT_TBL_NAME), Audit Program Type (AUDIT_PGM_TYPE_FLG), Key Validation (KEY_VALIDATION_FLG) and Caching strategy (CACHE_FLG) on MD Table (CI_MD_TBL). Override Label (OVRD_LABEL) and Customer Specific Description (DESCRLONG_OVRD) on Field Table.

Meta Data - Constraints

Properties	Description
Tables	CI_MD_CONST, CI_MD_CONST_FLD
Standard Data Fields	Constraint Id (CONST_ID) • Index Name for Primary Constraints • <Index Name>Rnn for Foreign Key Constraints Where • nn: integer, 01 through 99
Customer Modification	None

Meta Data - Menu

Menus can be extended to support multiple products by adding a new menu line to an existing menu. The sequence number on the menu line language table

(CI_MD_MENU_LINE_L) determines the order the menu lines appear. Within the same sequence, alphabetic sorting is used.

Properties	Description
Tables	CI_MD_MENU, CI_MD_MENU_L, CI_MD_MENU_ITEM, CI_MD_MENU_ITEM_L, CI_MD_MENU_LINE, CI_MD_MENU_LINE_L
Standard Data Fields	Menu Name (MENU_NAME), Menu Item Id (MENU_ITEM_ID), Menu Line Id (MENU_LINE_ID)
Customer Modification	Override Label (OVRD_LABEL) on Menu Line Language Table (CI_MD_MENU_LINE_L)

Meta Data - Program, Location and Services

Properties	Description
Tables	CI_MD_PRG_COM, CI_MD_PRG_LOC, CI_MD_SVC, CI_MD_SVC_L, CI_MD_SVC_PRG, CI_MD_PRG_MOD, CI_MD_PRG_EL_AT, CI_MD_PRG_ELEM, CI_MD_PRG_SEC, CI_MD_PRG_SQL, CI_MD_PRG_VAR, CI_MD_PRG_TAB
Standard Data Fields	Program Component Id (PROG_COM_ID), Location Id (LOC_ID), Program Component Name (PROG_COM_NAME), Service Name (SVC_NAME), Navigation Key (NAVIGATION_KEY)
Customer Modification	User Exit Program Name (USER_EXIT_PGM_NAME) on Program Components Table (CI_MD_PRG_COM),

Meta Data - Maintenance Object

Properties	Description
Tables	CI_MD_MO, CI_MD_MO_L, CI_MD_MO_TBL, CI_MD_MO_OPT, CI_MD_MO_ALG
Standard Data Fields	Maintenance Object (MAINT_OBJ_CD)
Customer Modification	None

Meta Data - Work Tables

Properties	Description
Tables	CI_MD_WRK_TBL, CI_MD_WRK_TBL_L, CI_MD_WRK_TBLFLD, CI_MD_MO_WRK
Standard Data Fields	Work Table Name (WRK_TBL_NAME)
Customer Modification	None

Meta Data - Search Object

Properties	Description
Tables	CI_MD_SO, CI_MD_SO_L, CI_MD_SO_RSFLD, CI_MD_SO_RSFLDAT, CI_MD_SOCG, CI_MD_SOCG_FLD, CI_MD_SOCG_FLDAT, CI_MD_SOCG_L, CI_MD_SOCG_SORT
Standard Data Fields	Search Object (SO_CD)
Customer Modification	None

Navigation Option

Properties	Description
Tables	CI_NAV_OPT, CI_NAV_OPT_L, CI_NAV_OPT_CTXT, CI_NAV_OPT_USG, CI_MD_NAV
Standard Data Fields	Navigation Option Code (NAV_OPT_CD), Navigation Key (NAVIGATION_KEY)
Customer Modification	None

Portal and Zone

Properties	Description
Tables	CI_PORTAL, CI_PORTAL_L, CI_PORTAL_ZONE, CI_PORTAL_OPT, CI_ZONE, CI_ZONE_L, CI_ZONE_PRM, CI_ZONE_HDL, CI_ZONE_HDL_L, CI_ZONE_HDL_PRM, CI_ZONE_HDL_PRM_L, CI_UI_ZONE
Standard Data Fields	Portal Code (PORTAL_CD), Zone Code (ZONE_CD), Zone Type Code (ZONE_HDL_CD) • A new Zone can be added to the Product owned Portal Pages. • The existing Zones cannot be removed from the Product owned Portal Pages.
Customer Modification	Sort Sequence (SORT_SEQ) on Context Sensitive Zone Table (CI_UI_ZONE). Show on Portal Preferences (USER_CONFIG_FLG) on Portal Table (CI_PORTAL). Override Sequence (SORT_SEQ_OVRD) on Portal Zone Table (CI_PORTAL_ZONE). Customer Specific Description (DESCRLONG_OVRD) on Zone Language Table (CI_ZONE_L). Override Parameter Value (ZONE_HDL_PARM_OVRD) on Zone Type Parameters Table (CI_ZONE_HDL_PRM). Override Parameter Value (ZONE_PARM_VAL_OVRD) on Zone Parameters Table (CI_ZONE_PRM).

Sequence

Properties	Description
Tables	CI_SEQ
Standard Data Fields	Sequence Name (SEQ_NAME)
Customer Modification	Sequence Number (SEQ_NBR) This field is updated by the application process and must be set to 1 initially.

Schema

Properties	Description
Tables	F1_SCHEMA
Standard Data Fields	Schema Name (SCHEMA_NAME)
Customer Modification	None

Script

Properties	Description
Tables	CI_SCR, CI_SCR_L, CI_SCR_CRT, CI_SCR_CRT_GRP, CI_SCR_CRT_GRP_L, CI_SCR_DA, CI_SCR_FLD_MAP, CI_SCR_PRMPT, CI_SCR_PRMPT_L, CI_SCR_STEP, CI_SCR_STEP_L
Standard Data Fields	Script (SCR_CD)
Customer Modification	None

To Do Type

Properties	Description
Tables	CI_TD_TYPE, CI_TD_TYPE_L, CI_TD_SRTKEY_TY, CI_TD_DRLKEY_TY, CI_TD_SRTKEY_TY_L
Standard Data Fields	To Do Type Code (TD_TYPE_CD)
Customer Modification	Creation Batch Code (CRE_BATCH_CD), Route Batch Code (RTE_BATCH_CD), Priority Flag (TD_PRIORITY_FLG) on To Do Type Table (CI_TD_TYPE)

XAI Configuration

Properties	Description
Tables	CI_XAI_ADAPTER, CI_XAI_ADAPTER_L, CI_XAI_CLASS, CI_XAI_CLASS_L, CI_XAI_ENV_HNDL, CI_XAI_ENV_HNDL_L, CI_XAI_FORMAT, CI_XAI_FORMAT_L, CI_XAI_RCVR, CI_XAI_RCVR_L, CI_XAI_RCVR_CTX, CI_XAI_RCVR_RSP, CI_XAI_RCVR_RGRP, CI_XAI_SENDER, CI_XAI_SERNDER_L, CI_XAI_SNDR_CTX, CI_XAI_OPTION
Standard Data Fields	Adapter Id (XAI_ADAPTER_ID), Class Id (XAI_CLASS_ID), Envelope Handler Id (XAI_ENV_HNDL_ID), XAI Format Id (XAI_FORMAT_ID), Receiver Id (XAI_RCVR_ID), Sender Id (XAI_SENDER_ID)
Customer Modification	Option Value (OPTION_VALUE) on Message Option Table (CI_XAI_OPTION)

XAI Services

Properties	Description
Tables	CI_XAI_IN_SVC, CI_XAI_IN_SVC_L, CI_XAI_SVC_PARM
Standard Data Fields	XAI Inbound Service Id (XAI_IN_SVC_ID), XAI Inbound Service Name (XAI_IN_SVC_NAME)
Customer Modification	XAI Version (XAI_VERSION_ID), Trace (TRACE_SW), Debug (DEBUG_SW), Request XSL (INPUT_XSL), Response XSL (RESPONSE_XSL), Record XSL (RECORD_XSL and Post Error (POST_ERROR_SW) on XAI Inbound Service Table (CI_XAI_IN_SVC)

Oracle Utilities Application Framework Only Tables

All data of the tables in this group belong to the Oracle Utilities Application Framework. No data modification or addition is allowed for these tables by base product development and customer modification. When an environment is upgraded to the next release of the Oracle Utilities Application Framework, the upgrade process will refresh the data in these tables.

- CI_MD_AT_DTL / CI_MD_AT_DTL_L

- CI_MD_ATT_TY

- CI_MD_CTL / CI_MD_CTL_L

- CI_MD_CTL_TMPL

- CI_MD_ELTY / CI_MD_ELTY_L

- CI_MD_ELTY_AT

- CI_MD_LOOKUP_F

- CI_MD_PDF / CI_MD_PDF_VAL

- CI_MD_MSG / CI_MD_MSG_L

- CI_MD_SRC_TYPE / CI_MD_SRC_TYPE_L

- CI_MD_TMPL / CI_MD_TMPL_L

- CI_MD_TMPL_ELTY

- CI_MD_TMPL_VAR / CI_MD_TMPL_VAR_L

- CI_MD_VAR / CI_MD_VAR_DTL / CI_MD_VAR_DTL_L

- CI_XAI_EXECUTER / CI_XAI_EXECUTER_L

System Table List

This section contains names of system tables, upgrade actions, and a brief description of tables. The upgrade actions are explained below.

Keep (KP): The data in the table in the customer's database is kept untouched. No insert or delete is performed to this table by the upgrade process. The initial installation will add necessary data for the system

Merge (MG): The non-base product data in the table in the database is kept untouched. If the data belongs to the base product, any changes pertaining to the new version of the software are performed.

Refresh (RF): The existing data in the table is replaced with the data from the base product table.

> **Note.** New product data is also inserted into tables marked as 'Merge'. If implementers add rows for a customer specific enhancement, it can cause duplication when the system data gets upgraded to the next version. We strongly recommend following the guidelines on how to use designated range of values or prefixes to segregate the implementation data from the base product data.

Table Name	Upgrade Action	Description
CI_ALG	MG	Algorithm
CI_ALG_L	MG	Algorithm Language
CI_ALG_PARM	MG	Algorithm Parameters
CI_ALG_TYPE	MG	Algorithm Type
CI_ALG_TYPE_L	MG	Algorithm Type Language
CI_ALG_TYPE_PRM	MG	Algorithm Type Parameter
CI_ALG_TYPE_PRM_L	MG	Algorithm Type Parameter Language
CI_ALG_VER	MG	Algorithm Version
CI_APP_SVC_ACC	MG	Application Service Access Mode
CI_BATCH_CTRL	MG	Batch Control
CI_BATCH_CTRL_ALG	MG	Batch Control Algorithm
CI_BATCH_CTRL_L	MG	Batch Control Language
CI_BATCH_CTRL_P	MG	Batch Control Parameters
CI_BATCH_CTRL_P_L	MG	Batch Control Parameters Language
CI_CHAR_ENTITY	MG	Characteristic Type Entity
CI_CHAR_TYPE	MG	Characteristic Type
CI_CHAR_TYPE_L	MG	Characteristic Type Language
CI_CHAR_VAL	MG	Characteristic Type Value
CI_CHAR_VAL_L	MG	Characteristic Type Value Language

Table Name	Upgrade Action	Description
CI_DISP_ICON	MG	Display Icon
CI_DISP_ICON_L	MG	Display Icon Language
CI_FK_REF	MG	Foreign Key Reference
CI_FK_REF_L	MG	Foreign Key Reference Language
CI_LANGUAGE	MG	Language Code
CI_LOOKUP_FIELD	MG	Lookup Field
CI_LOOKUP_VAL	MG	Lookup Field Value
CI_LOOKUP_VAL_L	MG	Lookup Field Value Language
CI_MD_CONST	MG	Constraints
CI_MD_CONST_FLD	MG	Constraint Fields
CI_MD_FLD	MG	Field
CI_MD_FLD_L	MG	Field Language
CI_MD_MENU	MG	Menu Information
CI_MD_MENU_IMOD	MG	Menu Item Module Maint
CI_MD_MENU_ITEM	MG	Menu Item
CI_MD_MENU_ITEM_L	MG	Menu Item Language
CI_MD_MENU_L	MG	Menu Language
CI_MD_MENU_LINE	MG	Menu Line
CI_MD_MENU_LINE_L	MG	Menu Line Language
CI_MD_MENU_MOD	MG	Menu Product Components
CI_MD_MO	MG	Maintenance Object
CI_MD_MO_ALG	MG	Maintenance Object Algorithm
CI_MD_MO_L	MG	Maintenance Object Language
CI_MD_MO_OPT	MG	Maintenance Object Option
CI_MD_MO_TBL	MG	Maintenance Object Table
CI_MD_MO_WRK	MG	Maintenance Object Work Tables
CI_MD_NAV	MG	Navigation Key
CI_MD_PRG_COM	MG	Program Components
CI_MD_PRG_ELEM	MG	UI Page Elements
CI_MD_PRG_EL_AT	MG	UI Page Element Attributes
CI_MD_PRG_LOC	MG	Program Location
CI_MD_PRG_MOD	MG	Program Module

Table Name	Upgrade Action	Description
CI_MD_PRG_SEC	MG	UI Page Sections
CI_MD_PRG_SQL	MG	MD SQL Meta Data
CI_MD_PRG_TAB	MG	UI Tab Meta Data
CI_MD_PRG_VAR	MG	Program Variable
CI_MD_SO	MG	Search Object
CI_MD_SOCG	MG	Search Object Criteria Group
CI_MD_SOCG_FLD	MG	Search Object Criteria Group Field
CI_MD_SOCG_FLDAT	MG	Search Criteria Group Field Attribute
CI_MD_SOCG_L	MG	Search Object Criteria Group Language
CI_MD_SOCG_SORT	MG	Search Criteria Group Result Sort Order
CI_MD_SO_L	MG	Search Object Language
CI_MD_SO_RSFLD	MG	Search Object Result Field
CI_MD_SO_RSFLDAT	MG	Search Object Result Field Attribute
CI_MD_SVC	MG	MD Service
CI_MD_SVC_L	MG	MD Service Language
CI_MD_SVC_PRG	MG	MD Service Program
CI_MD_TAB_MOD	MG	UI Tab Module
CI_MD_TBL	MG	MD Table
CI_MD_TBL_FLD	MG	MD Table Field
CI_MD_TBL_FLD_L	MG	MD Table Field Language
CI_MD_TBL_L	MG	MD Table Language
CI_MD_WRK_TBL	MG	Work Table
CI_MD_WRK_TBLFLD	MG	Work Table Field
CI_MD_WRK_TBL_L	MG	Work Table Language
CI_MSG	MG	Message
CI_MSG_CATEGORY	MG	Message Category
CI_MSG_CATEGORY_L	MG	Message Category Language
CI_MSG_L	MG	Message Language
CI_NAV_OPT	MG	Navigation Option
CI_NAV_OPT_CTXT	MG	Navigation Option Context
CI_NAV_OPT_L	MG	Navigation Option Language
CI_NAV_OPT_USG	MG	Navigation Option Usage

Table Name	Upgrade Action	Description
CI_PORTAL	MG	Portal
CI_PORTAL_L	MG	Portal Language
C1_PORTAL_OPT	MG	Portal Option
CI_PORTAL_ZONE	MG	Portal Zone
CI_SCR	MG	Script
CI_SCR_CRT	MG	Script Criteria
CI_SCR_CRT_GRP	MG	Script Criteria Group
CI_SCR_CRT_GRP_L	MG	Script Criteria Group Language
CI_SCR_DA	MG	Script Data Area
CI_SCR_FLD_MAP	MG	Script Field Mapping
CI_SCR_L	MG	Script Language
CI_SCR_PRMPT	MG	Script Prompt
CI_SCR_PRMPT_L	MG	Script Prompt Language
CI_SCR_STEP	MG	Script Step
CI_SCR_STEP_L	MG	Script Step Language
CI_SEQ	MG	Sequence
CI_TD_DRLKEY_TY	MG	To Do Type Drill Key
CI_TD_SRTKEY_TY	MG	To Do Type Sort Key
CI_TD_SRTKEY_TY_L	MG	To Do Type Sort Key Language
CI_TD_TYPE	MG	To Do Type
CI_TD_TYPE_L	MG	To Do Type Language
CI_UI_ZONE	MG	Context Sensitive Zone
CI_USR_NAV_LINK	MG	User Favorite Links
CI_XAI_ADAPTER	MG	XAI Adapter
CI_XAI_ADAPTER_L	MG	XAI Adapter Lang
CI_XAI_CLASS	MG	Message Class
CI_XAI_CLASS_L	MG	Message Class Language
CI_XAI_ENV_HNDL	MG	XAI Envelope Handler
CI_XAI_ENV_HNDL_L	MG	XAI Envelope Handler Language
CI_XAI_IN_SVC	MG	XAI Inbound Service
CI_XAI_IN_SVC_L	MG	XAI Inbound Service Language
CI_XAI_SVC_PARM	MG	XAI Inbound Service Parameters

Table Name	Upgrade Action	Description
CI_ZONE	MG	Zone
CI_ZONE_HDL	MG	Zone Type
CI_ZONE_HDL_L	MG	Zone Type Language
CI_ZONE_HDL_PRM	MG	Zone Type Parameters
CI_ZONE_HDL_PRM_L	MG	Zone Type Parameters Language
CI_ZONE_L	MG	Zone Language
CI_ZONE_PRM	MG	Zone Parameters
F1_BUS_OBJ	MG	Business Object
F1_BUS_OBJ_ALG	MG	Business Object Algorithm
F1_BUS_OBJ_L	MG	Business Object Language
F1_BUS_OBJ_OPT	MG	Business Object Option
F1_BUS_OBJ_STATUS	MG	Business Object Status
F1_BUS_OBJ_STATUS_ALG	MG	Business Object Status Algorithm
F1_BUS_OBJ_STATUS_L	MG	Business Object Status Language
F1_BUS_OBJ_STATUS_OPT	MG	Business Object Status Option
F1_BUS_OBJ_STATUS_RSN	MG	Status Reason
F1_BUS_OBJ_STATUS_RSN_L	MG	Status Reason Language
F1_BUS_OBJ_TR_RULE	MG	Business Object Transition Rule
F1_BUS_OBJ_TR_RULE_L	MG	Business Object Transition Rule Language
F1_BUS_SVC	MG	Business Service
F1_BUS_SVC_L	MG	Business Service Language
F1_DATA_AREA	MG	Data Area
F1_DATA_AREA_L	MG	Data Area Language
F1_DB_OBJECTS_REPO	MG	Database Objects Repository
F1_EXT_LOOKUP_VAL	MG	Extendable Lookup
F1_EXT_LOOKUP_VAL_L	MG	Extendable Lookup Language
F1_EXT_LOOKUP_VAL_CHAR	MG	Extendable Lookup Characteristics
F1_IWS_ANN	MG	Web Service Annotation
F1_IWS_ANN_L	MG	Web Service Annotation Language
F1_IWS_ANN_PARM	MG	Web Service Annotation Parameter
F1_IWS_ANN_TYPE	MG	Web Service Annotation Type

Table Name	Upgrade Action	Description
F1_IWS_ANN_TYPE_L	MG	Web Service Annotation Type Language
F1_IWS_ANN_TYPE_PARM	MG	Web Service Annotation Type Parm
F1_IWS_ANN_TYPE_PARM_L	MG	Web Service Annotation Type Parameter Language
F1_IWS_SVC	MG	Inbound Web Service
F1_IWS_SVC_L	MG	Inbound Web Service Language
F1_IWS_SVC_OPER	MG	Inbound Web Service Operations
F1_IWS_SVC_OPER_L	MG	Inbound Web Service Operations Language
F1_MANAG_CONTENT	MG	Managed Content
F1_MANAG_CONTENT_L	MG	Managed Content Language
F1_MAP	MG	UI Map
F1_MAP_L	MG	UI Map Language
F1_MIGR_PLAN	MG	Migration Plan
F1_MIGR_PLAN_INSTR	MG	Migration Plan Instruction
F1_MIGR_PLAN_INSTR_ALG	MG	Migration Plan Instruction Algorithm
F1_MIGR_PLAN_INSTR_L	MG	Migration Plan Instruction Language
F1_MIGR_PLAN_L	MG	Migration Plan Language
F1_MIGR_REQ	MG	Migration Request
F1_MIGR_REQ_INCL_REQ	MG	Migration Request Grouping
F1_MIGR_REQ_INSTR	MG	Migration Request Instruction
F1_MIGR_REQ_INSTR_ENTITY	MG	Migration Request Instruction Entity
F1_MIGR_REQ_INSTR_L	MG	Migration Request Instruction Language
F1_MIGR_REQ_L	MG	Migration Request Language
F1_MIGR_REQ_INCL_REQ	MG	Migration Request Grouping
F1_SCHEMA	MG	Schema
SC_ACCESS_CNTL	MG	User Group Access Control
SC_APP_SERVICE	MG	Application Service
SC_APP_SERVICE_L	MG	Application Service Language
SC_USR_GRP_PROF	MG	User Group Profile
CI_ACC_GRP	KP	Access Group

Table Name	Upgrade Action	Description
CI_ACC_GRP_DAR	KP	Access Group / Data Access Group
CI_ACC_GRP_L	KP	Access Group Language
CI_APP_SVC_SCTY	KP	Security Type Application Service
CI_CAL_HOL	KP	Work Calendar Holidays
CI_CAL_HOL_L	KP	Work Calendar Holidays Language
CI_CAL_WORK	KP	Work Calendar
CI_CAL_WORK_L	KP	Work Calendar Language
CI_CHTY_TDTY	KP	To Do Type Template Characteristics
CI_COUNTRY	KP	Country
CI_COUNTRY_L	KP	Country Language
CI_CURRENCY_CD	KP	Currency Code
CI_CURRENCY_CD_L	KP	Currency Code Language
CI_DAR	KP	Data Access Role
CI_DAR_L	KP	Data Access Language
CI_DAR_USR	KP	Data Access User
CI_DISP_PROF	KP	Display Profile
CI_DISP_PROF_L	KP	Display Profile Language
CI_FUNC	KP	Function
CI_FUNC_FLD	KP	Function Field
CI_FUNC_FLD_L	KP	Function Field Language
CI_FUNC_L	KP	Function Language
CI_GEO_TYPE	KP	Geographic Type
CI_GEO_TYPE_L	KP	Geographic Type Language
CI_INSTALL_ALG	KP	Installation Algorithm
CI_INSTALL_MSG	KP	Installation Message
CI_INSTALL_MSG_L	KP	Installation Message Language
CI_INSTALL_PROD	KP	Installation Product
CI_MD_RPT	KP	Report Definition
CI_MD_RPT_L	KP	Report Language
CI_MD_RPT_LBL	KP	Report Labels
CI_MD_RPT_PARM	KP	Report Parameters

Table Name	Upgrade Action	Description
CI_MD_RPT_PARM_L	KP	Report Parameters Language
CI_MD_TOOLREP_XML	KP	MD Tool Reference XML
CI_MD_TOOL_REP	KP	MD Tool Reference
CI_NT_DNTY_CTXT	KP	Notification Download Type Context
CI_NT_DWN_FORM	KP	Notification Download Format
CI_NT_DWN_FORM_L	KP	Notification Download Format Language
CI_NT_DWN_PROF	KP	Notification Download Profile
CI_NT_DWN_PROF_L	KP	Notification Download Profile Language
CI_NT_DWN_TYPE	KP	Notification Download Type
CI_NT_DWN_TYPE_L	KP	Notification Download Type Language
CI_NT_UP_XTYPE	KP	Notification Upload Type
CI_NT_UP_XTYPE_L	KP	Notification Upload Type Language
CI_NT_XID	KP	External System
CI_NT_XID_L	KP	External System Language
CI_PHONE_TYPE	KP	Phone Type
CI_PHONE_TYPE_L	KP	Phone Type Language
CI_ROLE	KP	Role
CI_ROLE_L	KP	Role Language
CI_ROLE_USER	KP	Role User
CI_RPT_OPTION	KP	Report Options
CI_SC_AUTH_LVL	KP	Security Type Auth Level
CI_SC_AUTH_LVL_L	KP	Security Type Auth Level Language
CI_SC_TYPE	KP	Security Type
CI_SC_TYPE_L	KP	Security Type Language
CI_SEAS_SHIFT	KP	Seasonal Time Shift Schedule
CI_SEAS_TM_SHIFT	KP	Seasonal Time Shift
CI_SEAS_TM_SHIFT_L	KP	Seasonal Shift Language
CI_STATE	KP	State
CI_STATE_L	KP	State Language
CI_TD_EX_LIST	KP	To Do Type Message Overrides
CI_TD_TYPE_ALG	KP	To Do Type Algorithms

Table Name	Upgrade Action	Description
CI_TD_TYPE_CHAR	KP	To Do Type Characteristic
CI_TD_VAL_ROLE	KP	To Do Type Role
CI_TIME_ZONE	KP	Time Zone
CI_TIME_ZONE_L	KP	Time Zone Language
CI_USR_GRP_SC	KP	User Group Security Type
CI_USR_BOOKMARK	KP	User Bookmarks
CI_USR_PORTAL	KP	User Portal
CI_USR_SCR	KP	User Scripts
CI_USR_ZONE	KP	User Zone
CI_USR_ZONE_SAVE	KP	User Zone Save
CI_WFM	KP	Feature Configuration
CI_WFM_L	KP	Feature Configuration Language
CI_WFM_MSG	KP	Feature Configuration Message
CI_WFM_OPT	KP	Feature Configuration Options
CI_WF_EVT_TYPE	KP	WF Event Type
CI_WF_EVT_TYPE_L	KP	WF Event Type Language
CI_WF_PP	KP	WF Process Profile
CI_WF_PP_L	KP	WF Process Profile Language
CI_WF_PP_NT	KP	WF Process Notification
CI_WF_PP_NT_CRT	KP	WF Process Notification Criteria
CI_WF_PROC_SCHED	KP	WF Process Creation Schedule
CI_WF_PROC_SCHED_K	KP	WF Process Creation Schedule Key
CI_WF_PROC_TMPL	KP	WF Process Template
CI_WF_PROC_TMPL_L	KP	WF Process Template Language
CI_WF_RESP	KP	WF Response
CI_WF_RESP_DEP	KP	WF Response Dependency
CI_XAI_JDBC_CON	KP	XAI JDBC Connection
CI_XAI_JDBC_CON_L	KP	XAI JDBC Connection Language
CI_XAI_JMS_CON	KP	XAI JMS Connection
CI_XAI_JMS_CON_L	KP	XAI JMS Connection Language
CI_XAI_JMS_Q	KP	XAI JMS Queue
CI_XAI_JMS_Q_L	KP	XAI JMS Queue Language

Table Name	Upgrade Action	Description
CI_XAI_JMS_TPC	KP	XAI JMS Topic
CI_XAI_JMS_TPC_L	KP	XAI JMS Topic Language
CI_XAI_JNDI_SVR	KP	XAI JNDI Server
CI_XAI_JNDI_SVR_L	KP	XAI JNDI Server Language
CI_XAI_OPTION	KP	Message Option
CI_XAI_RCVR	KP	XAI Receiver
CI_XAI_RCVR_CTX	KP	XAI Receiver Context
CI_XAI_RCVR_L	KP	XAI Receiver Language
CI_XAI_RCVR_RGRP	KP	XAI Receiver Rule Group
CI_XAI_RCVR_RSP	KP	XAI Receiver Response
CI_XAI_RGRP	KP	XAI Rule Group
CI_XAI_RGRP_ATT	KP	XAI Rule Group Attachment
CI_XAI_RGRP_L	KP	XAI Rule Group Language
CI_XAI_ROUTING	KP	XAI Routing
CI_XAI_RT_TYPE	KP	XAI Route Type
CI_XAI_RT_TYPE_L	KP	XAI Route Type Language
CI_XAI_RULE	KP	XAI Rule
CI_XAI_SENDER	KP	Message Sender
CI_XAI_SENDER_L	KP	Message Sender Language
CI_XAI_SNDR_CTX	KP	Message Sender Context
F1_BKT_CONFIG	KP	Bucket Configuration
F1_BKT_CONFIG_L	KP	Bucket Configuration Language
F1_BKT_CONFIG_REL_OBJ	KP	Bucket Configuration Related Object
F1_BKT_CONFIG_VAL	KP	Bucket Configuration Value
F1_BKT_CONFIG_VAL_L	KP	Bucket Configuration Value Language
F1_BUS_OBJ_STATUS_RSN_ CHAR	KP	Status Reason Characteristic
F1_EXTSYS_OUTMSG_PROF	KP	External System Outbound Message Type
F1_INSTALLATION	KP	Installation Option - Framework
F1_IWS_ANN_CHAR	KP	Web Service Annotation Characteristics
F1_IWS_ANN_TYPE_CHAR	KP	Web Service Annotation Type Characteristics

Table Name	Upgrade Action	Description
F1_IWS_SVC_ANN	KP	Inbound Web Service Link to Annotation
F1_IWS_SVC_CHAR	KP	Inbound Web Service Characteristics
F1_IWS_SVC_LOG	KP	Inbound Web Service Log
F1_IWS_SVC_LOG_PARM	KP	Inbound Web Service Log Parameter
F1_MAP_OVRD	KP	UI Map Override
F1_MD_DB_OBJ	KP	MD Database Object
F1_MST_CONFIG	KP	Master Configuration
F1_OUTMSG_TYPE	KP	Outbound Message Type
F1_OUTMSG_TYPE_L	KP	Outbound Message Type Language
F1_REQ_TYPE	KP	Request Type
F1_REQ_TYPE_L	KP	Request Type Language
F1_REQ_TYPE_LOG	KP	Request Type Log
F1_REQ_TYPE_LOG_PARM	KP	Request Type Log Parameters
F1_SVC_TASK_TYPE	KP	Service Task Type
F1_SVC_TASK_TYPE_CHAR	KP	Service Task Type Characteristics
F1_SVC_TASK_TYPE_L	KP	Service Task Type Language
F1_WEB_SVC	KP	Web Service Adapter
F1_WEB_SVC_CHAR	KP	Web Service Adapter Characteristics
F1_WEB_SVC_L	KP	Web Service Adapter Language
F1_WEB_SVC_LOG	KP	Web Service Adapter Log
F1_WEB_SVC_LOG_PARM	KP	Web Service Adapter Log Parameter
F1_WEB_SVC_OPERATIONS	KP	Web Service Adapter Operations
SC_USER	KP	User
SC_USER_CHAR	KP	User Characteristic
SC_USER_GROUP	KP	User Group
SC_USER_GROUP_L	KP	User Group Language
SC_USR_GRP_USR	KP	User Group User
CI_MD_ATT_TY	RF	MD Element Attribute Type
CI_MD_AT_DTL	RF	MD Element Attribute Type Detail
CI_MD_AT_DTL_L	RF	MD Element Attribute Type Detail Language
CI_MD_CTL	RF	Generator Control

Table Name	Upgrade Action	Description
CI_MD_CTL_L	RF	Generator Control Language
CI_MD_CTL_TMPL	RF	Generator Control Template
CI_MD_ELTY	RF	MD Element Type
CI_MD_ELTY_AT	RF	Element Type Attributes
CI_MD_ELTY_L	RF	Element Type Language
CI_MD_LOOKUP_F	RF	MD Lookup Field
CI_MD_MSG	RF	MD Message
CI_MD_MSG_L	RF	MD Message Language
CI_MD_PDF	RF	Predefined Fields
CI_MD_PDF_VAL	RF	Predefined Values
CI_MD_SRC_TYPE	RF	Source Type
CI_MD_SRC_TYPE_L	RF	Source Type Language
CI_MD_TMPL	RF	Template
CI_MD_TMPL_ELTY	RF	Template Element Types
CI_MD_TMPL_L	RF	Template Language
CI_MD_TMPL_VAR	RF	Template Variable
CI_MD_TMPL_VAR_L	RF	Template Variable Language
CI_MD_VAR	RF	Variable
CI_MD_VAR_DTL	RF	Variable Detail
CI_MD_VAR_DTL_L	RF	Variable Detail Language
CI_XAI_EXECUTER	RF	XAI Executer
CI_XAI_EXECUTER_L	RF	XAI Executer Language

Appendix D

Partitioning Recommendations

This section specifies the partitioning and compression strategies recommended for an initial database configuration. It includes the following topics:

- **Partitioning Recommendations**

Partitioning Recommendations

In general, the recommendation is for a minimum of 'n' partitions for selective database objects, where 'n' is number of RAC nodes. The specific table level partitioning recommendations are as follows:

- M1_GPS_DATA table can be partitioned by CAP_DTTM (smaller partition better performance). Begin with bi-weekly partitions.

- The Indexes need to Global Indexes as the partitioning key CAP_DTTM is not a part of the Index.

The following sections provide partition recommendation and can be used as a reference. It includes the following:

- **M1_GPS_DATA**

- **M1_CREW_SHFT**

- **M1_CREW_SHFT_CHAR**

- **M1_CREW_SHFT_DEPOT**

- **M1_CREW_SHFT_LOC**

- **M1_CREW_SHFT_LOG**

- **M1_CREW_SHFT_MLOG_PARM**

- **M1_CREW_SHFT_MLOG**

- **M1_CREW_SHFT_MLOG_PARM**

- **M1_CREW_SHFT_RESRC**

- **M1_CREW_SHFT_SVC_AREA**

- **M1_CREW_SHFT_SVC_CLS**

M1_GPS_DATA

```
CREATE TABLE M1_GPS_DATA
  (
    GPS_DATA_ID CHAR(14 BYTE) NOT NULL ENABLE,
```

```
            CAP_DTTM DATE NOT NULL ENABLE,
            RESRC_ID        CHAR(12 BYTE) DEFAULT ' ' NOT NULL ENABLE,
            EXT_SRC_TYPE_FLG CHAR(4 BYTE) DEFAULT ' ',
            EXT_SRC_ID       VARCHAR2(254 BYTE) DEFAULT ' ',
            USER_ID         CHAR(8 BYTE) DEFAULT ' ' NOT NULL ENABLE,
            GEO_LAT         NUMBER(9,6),
            GEO_LONG        NUMBER(9,6),
            GEO_ALT         NUMBER(9,6) DEFAULT 0 NOT NULL ENABLE,
            SPEED           NUMBER(9,6) DEFAULT 0 NOT NULL ENABLE,
            TRK_ANGLE       NUMBER(9,6) DEFAULT 0 NOT NULL ENABLE,
            VERSION         NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
          M1_GEOM MDSYS.SDO_GEOMETRY
          )
    PARTITION BY RANGE
      (CAP_DTTM )
    (PARTITION P1 VALUES LESS THAN(TO_DATE('01/01/2011 00:00:01','DD/MM/
    YYYY HH24:MI:SS')),
     PARTITION P2 VALUES LESS THAN(TO_DATE('15/01/2011 00:00:01','DD/MM/
    YYYY HH24:MI:SS')),
     PARTITION P3 VALUES LESS THAN(TO_DATE('01/02/2011 00:00:01','DD/MM/
    YYYY HH24:MI:SS')),
     PARTITION P4 VALUES LESS THAN(TO_DATE('15/02/2011 00:00:01','DD/MM/
    YYYY HH24:MI:SS')),
     PARTITION P5 VALUES LESS THAN(TO_DATE('01/03/2011 00:00:01','DD/MM/
    YYYY HH24:MI:SS')),
        PARTITION P6 VALUES LESS THAN (MAXVALUE) )
    /

    CREATE UNIQUE INDEX "CISADM"."MT315P0" ON "CISADM"."M1_GPS_DATA"
      (  "GPS_DATA_ID" )
    /

     CREATE INDEX "CISADM"."MT315S1_TEST" ON "CISADM"."M1_GPS_DATA"
        (  "RESRC_ID",  "CAP_DTTM"  )
    /

    CREATE INDEX "CISADM"."MT315S2" ON "CISADM"."M1_GPS_DATA"
        ( "CAP_DTTM" )
    /
```

M1_CREW_SHFT

```
    CREATE TABLE M1_CREW_SHFT
      (
        SUBSCRIPTION_ID    CHAR(14 BYTE) DEFAULT ' ' NOT NULL ENABLE,
        USER_ID           CHAR(8 BYTE),
        MDT_ID            CHAR(14 BYTE),
        LOGON_MOBILE_PHONE VARCHAR2(24 BYTE),
        SHFT_WKTMPL_ID     CHAR(14 BYTE),
        FROM_WEEK_NBR      NUMBER(3,0),
        FROM_WEEK_DAY_FLG  CHAR(4 BYTE),
        FROM_TIME DATE,
        TO_WEEK_NBR NUMBER(3,0),
        TO_TIME DATE,
        BO_DATA_AREA CLOB,
        VERSION           NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
        TO_WEEK_DAY_FLG   CHAR(4 BYTE),
        TIME_ZONE_CD     CHAR(10 BYTE) DEFAULT ' ' NOT NULL ENABLE,
        M1_GPS_RESRC_ID  CHAR(12 BYTE) DEFAULT ' ',
        M1_SCHED_OPT_FLG CHAR(4 BYTE) DEFAULT ' ' NOT NULL ENABLE,
        M1_REQ_BY_FLG    CHAR(4 BYTE) DEFAULT ' ',
```

```
               M1_OFFLINE_FLG   CHAR(4 BYTE) DEFAULT ' ',
               CREW_SHFT_ID     CHAR(14 BYTE) NOT NULL ENABLE,
               BUS_OBJ_CD       CHAR(30 BYTE) NOT NULL ENABLE,
               BO_STATUS_CD     CHAR(12 BYTE),
               STATUS_UPD_DTTM DATE,
               BUS_STATUS_DTTM DATE,
               BO_STATUS_REASON_CD  VARCHAR2(30 BYTE),
               CREW_SHFT_TYPE_CD    VARCHAR2(30 BYTE) NOT NULL ENABLE,
               RESRC_ID             CHAR(12 BYTE),
               SHFT_COPRF_CD        VARCHAR2(30 BYTE) DEFAULT ' ',
               DRIP_MODE_FLG        CHAR(4 BYTE),
               DRIP_HORIZON         NUMBER(3,0),
               GPS_DATA_ENABLED_FLG CHAR(4 BYTE),
               SHFT_USAGE_FLG       CHAR(4 BYTE) DEFAULT ' ' NOT NULL ENABLE,
               PLAN_START_DTTM DATE,
               PLAN_END_DTTM DATE,
               SHFT_TMPL_ID   VARCHAR2(30 BYTE),
               M1_OVRD_FLG    CHAR(4 BYTE),
               M1_CAPACITY    NUMBER(4,0),
               TASK_TYPE_CD   VARCHAR2(30 BYTE),
               SHIFT_DURATION NUMBER(6,0),
               ORIG_PLANNED_START_DTTM DATE,
               ORIG_PLANNED_END_DTTM DATE,
               VARIABLE_SHIFT_FLG      CHAR(4 BYTE),
               M1_UNDISPATCH_OPTION_FLG CHAR(4 BYTE) DEFAULT ' ' NOT NULL ENABLE,
               M1_CHANGE_SEQUENCE_FLG   CHAR(4 BYTE) DEFAULT ' ' NOT NULL ENABLE,
               M1_AUTO_COMP_FLG        CHAR(4 BYTE),
               ILM_DT DATE,
               ILM_ARCH_SW CHAR(1 BYTE)
              ) TABLESPACE <Tablespace_Name>
ENABLE ROW MOVEMENT
LOB (BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS
CACHE)
PARTITION BY RANGE (ILM_DT)
SUBPARTITION BY range (CREW_SHFT_ID)
SUBPARTITION TEMPLATE(
subpartition SUB1 values less than (124999999999),
subpartition SUB2 values less than (249999999999),
subpartition SUB3 values less than (374999999999),
subpartition SUB4 values less than (499999999999),
subpartition SUB5 values less than (624999999999),
subpartition SUB6 values less than (744999999999),
subpartition SUB7 values less than (874999999999),
subpartition SUB8 values less than (maxvalue)
)
(
PARTITION P1 VALUES LESS THAN(TO_DATE('15/12/2010 00:00:01','DD/MM/
YYYY HH24:MI:SS')),
PARTITION P2 VALUES LESS THAN(TO_DATE('13/02/2011 00:00:01','DD/MM/
YYYY HH24:MI:SS')),
PARTITION P3 VALUES LESS THAN(TO_DATE('14/04/2011 00:00:01','DD/MM/
YYYY HH24:MI:SS')),
PARTITION P4 VALUES LESS THAN(TO_DATE('13/06/2011 00:00:01','DD/MM/
YYYY HH24:MI:SS')),
PARTITION P5 VALUES LESS THAN(TO_DATE('12/08/2011 00:00:01','DD/MM/
YYYY HH24:MI:SS')),
PARTITION P6 VALUES LESS THAN(TO_DATE('11/10/2011 00:00:01','DD/MM/
YYYY HH24:MI:SS')),
PARTITION P7 VALUES LESS THAN(TO_DATE('10/12/2011 00:00:01','DD/MM/
YYYY HH24:MI:SS')),
PARTITION P8 VALUES LESS THAN(MAXVALUE)
);
```

```
CREATE UNIQUE INDEX MT100P0 on M1_CREW_SHFT (CREW_SHFT_ID ) TABLESPACE
<Tablespace_Name> local ;
ALTER TABLE M1_CREW_SHFT ADD CONSTRAINT MT100P0 PRIMARY KEY
(CREW_SHFT_ID) USING INDEX TABLESPACE <Tablespace_Name>;
```

M1_CREW_SHFT_CHAR

```
CREATE TABLE M1_CREW_SHFT_CHAR
  (
    CREW_SHFT_ID    CHAR(14 BYTE) NOT NULL ENABLE,
    CHAR_TYPE_CD    CHAR(8 BYTE) NOT NULL ENABLE,
    SEQ_NUM         NUMBER(3,0) NOT NULL ENABLE,
    CHAR_VAL        CHAR(16 BYTE) DEFAULT '' NOT NULL ENABLE,
    ADHOC_CHAR_VAL CHAR(254 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK1    VARCHAR2(50 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK2    VARCHAR2(50 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK3    VARCHAR2(50 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK4    VARCHAR2(50 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK5    VARCHAR2(50 BYTE) DEFAULT '' NOT NULL ENABLE,
    SRCH_CHAR_VAL  VARCHAR2(50 BYTE),
    VERSION         NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    CONSTRAINT M1_CREW_SHFT_CHAR_FK FOREIGN KEY(CREW_SHFT_ID)
REFERENCES M1_CREW_SHFT
    )
PARTITION BY REFERENCE (M1_CREW_SHFT_CHAR_FK)
ENABLE ROW MOVEMENT;

CREATE UNIQUE INDEX MT208P0 ON M1_CREW_SHFT_CHAR (CREW_SHFT_ID,
CHAR_TYPE_CD, SEQ_NUM) TABLESPACE <Index Tablespace_Name>
GLOBAL PARTITION BY RANGE(CREW_SHFT_ID)
 (
 PARTITION PART1 VALUES LESS THAN ('124999999999'),
 PARTITION PART2 VALUES LESS THAN ('249999999999'),
 PARTITION PART3 VALUES LESS THAN ('374999999999'),
 PARTITION PART4 VALUES LESS THAN ('499999999999'),
 PARTITION PART5 VALUES LESS THAN ('624999999999'),
 PARTITION PART6 VALUES LESS THAN ('749999999999'),
 PARTITION PART7 VALUES LESS THAN ('874999999999'),
 PARTITION PART8 VALUES LESS THAN (MAXVALUE)
 );
ALTER TABLE M1_CREW_SHFT_CHAR ADD CONSTRAINT MT208P0 PRIMARY KEY
(CREW_SHFT_ID, CHAR_TYPE_CD, SEQ_NUM) USING INDEX;

CREATE INDEX MT208S1 ON M1_CREW_SHFT_CHAR(SRCH_CHAR_VAL)
GLOBAL PARTITION BY HASH(SRCH_CHAR_VAL)
 (
 PARTITION PART1 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART2 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART3 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART4 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART5 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART6 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART7 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART8 TABLESPACE <Index Tablespace_Name>
);
```

M1_CREW_SHFT_DEPOT

```
CREATE TABLE M1_CREW_SHFT_DEPOT
  (
```

```
        CREW_SHFT_ID CHAR(14 BYTE) NOT NULL ENABLE,
        M1_DEPOT_CD  VARCHAR2(30 BYTE) NOT NULL ENABLE,
        VERSION      NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
      CONSTRAINT M1_CREW_SHFT_DEPOT_FK FOREIGN KEY(CREW_SHFT_ID)
REFERENCES M1_CREW_SHFT
        )
PARTITION BY REFERENCE (M1_CREW_SHFT_DEPOT_FK)
ENABLE ROW MOVEMENT;


CREATE UNIQUE INDEX M1T829P0 ON M1_CREW_SHFT_DEPOT (CREW_SHFT_ID,
M1_DEPOT_CD) TABLESPACE <Index Tablespace_Name>
GLOBAL PARTITION BY RANGE(CREW_SHFT_ID)
 (
 PARTITION PART1 VALUES LESS THAN ('124999999999'),
 PARTITION PART2 VALUES LESS THAN ('249999999999'),
 PARTITION PART3 VALUES LESS THAN ('374999999999'),
 PARTITION PART4 VALUES LESS THAN ('499999999999'),
 PARTITION PART5 VALUES LESS THAN ('624999999999'),
 PARTITION PART6 VALUES LESS THAN ('749999999999'),
 PARTITION PART7 VALUES LESS THAN ('874999999999'),
 PARTITION PART8 VALUES LESS THAN (MAXVALUE)
 );


ALTER TABLE M1_CREW_SHFT_DEPOT ADD CONSTRAINT M1T829P0 PRIMARY KEY
(CREW_SHFT_ID, M1_DEPOT_CD) USING INDEX;
```

M1_CREW_SHFT_LOC

```
CREATE TABLE M1_CREW_SHFT_LOC
  (
    CREW_SHFT_ID CHAR(14 BYTE) NOT NULL ENABLE,
    SEQNO        NUMBER(5,0) NOT NULL ENABLE,
    LOC_USG_FLG  CHAR(4 BYTE) NOT NULL ENABLE,
    LOC_TYPE_FLG CHAR(4 BYTE) NOT NULL ENABLE,
    LOCATION_CD  VARCHAR2(30 BYTE),
    BO_DATA_AREA CLOB,
    VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    CONSTRAINT M1_CREW_SHFT_LOC_FK FOREIGN KEY(CREW_SHFT_ID)
REFERENCES M1_CREW_SHFT
    )
LOB (BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS
CACHE)
PARTITION BY REFERENCE (M1_CREW_SHFT_LOC_FK)
ENABLE ROW MOVEMENT;

CREATE UNIQUE INDEX MT201P0 ON M1_CREW_SHFT_LOC (CREW_SHFT_ID, SEQNO)
TABLESPACE <Index Tablespace_Name>
GLOBAL PARTITION BY RANGE(CREW_SHFT_ID)
 (
 PARTITION PART1 VALUES LESS THAN ('124999999999'),
 PARTITION PART2 VALUES LESS THAN ('249999999999'),
 PARTITION PART3 VALUES LESS THAN ('374999999999'),
 PARTITION PART4 VALUES LESS THAN ('499999999999'),
 PARTITION PART5 VALUES LESS THAN ('624999999999'),
 PARTITION PART6 VALUES LESS THAN ('749999999999'),
 PARTITION PART7 VALUES LESS THAN ('874999999999'),
 PARTITION PART8 VALUES LESS THAN (MAXVALUE)
 );


ALTER TABLE M1_CREW_SHFT_LOC ADD CONSTRAINT MT201P0 PRIMARY KEY
(CREW_SHFT_ID, SEQNO) USING INDEX;
```

```
CREATE INDEX MT201S1 ON M1_CREW_SHFT_LOC(LOCATION_CD)
GLOBAL PARTITION BY HASH(LOCATION_CD)
 (
 PARTITION PART1 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART2 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART3 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART4 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART5 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART6 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART7 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART8 TABLESPACE <Index Tablespace_Name>
);
```

M1_CREW_SHFT_LOG

```
CREATE TABLE M1_CREW_SHFT_LOG
  (
    CREW_SHFT_ID        CHAR(14 BYTE) NOT NULL ENABLE,
    SEQNO               NUMBER(5,0) NOT NULL ENABLE,
    BO_STATUS_REASON_CD VARCHAR2(30 BYTE),
    LOG_ENTRY_TYPE_FLG  CHAR(4 BYTE) DEFAULT '' NOT NULL ENABLE,
    BO_STATUS_CD        CHAR(12 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_TYPE_CD        CHAR(8 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL            CHAR(16 BYTE) DEFAULT '' NOT NULL ENABLE,
    ADHOC_CHAR_VAL      VARCHAR2(254 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK1        VARCHAR2(50 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK2        VARCHAR2(50 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK3        VARCHAR2(50 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK4        VARCHAR2(50 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK5        VARCHAR2(50 BYTE) DEFAULT '' NOT NULL ENABLE,
    DESCRLONG           VARCHAR2(4000 BYTE) DEFAULT '' NOT NULL ENABLE,
    LOG_DTTM DATE NOT NULL ENABLE,
    MESSAGE_CAT_NBR NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,
    MESSAGE_NBR     NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,
    USER_ID         CHAR(8 BYTE) DEFAULT '' NOT NULL ENABLE,
    VERSION         NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    BUS_STATUS_DTTM DATE,
    CONSTRAINT M1_CREW_SHFT_LOG_FK FOREIGN KEY(CREW_SHFT_ID)
REFERENCES M1_CREW_SHFT
   )
PARTITION BY REFERENCE (M1_CREW_SHFT_LOG_FK)
ENABLE ROW MOVEMENT;

CREATE UNIQUE INDEX MT204P0 ON M1_CREW_SHFT_LOG(CREW_SHFT_ID,SEQNO)
TABLESPACE <Index Tablespace_Name>
GLOBAL PARTITION BY RANGE(CREW_SHFT_ID)
 (
 PARTITION "PART1" VALUES LESS THAN ('124999999999'),
 PARTITION "PART2" VALUES LESS THAN ('249999999999'),
 PARTITION "PART3" VALUES LESS THAN ('374999999999'),
 PARTITION "PART4" VALUES LESS THAN ('499999999999'),
 PARTITION "PART5" VALUES LESS THAN ('624999999999'),
 PARTITION "PART6" VALUES LESS THAN ('749999999999'),
 PARTITION "PART7" VALUES LESS THAN ('874999999999'),
 PARTITION "PART8" VALUES LESS THAN (MAXVALUE)
 );

ALTER TABLE M1_CREW_SHFT_LOG ADD CONSTRAINT MT204P0 PRIMARY
KEY(CREW_SHFT_ID,SEQNO) USING INDEX;
```

M1_CREW_SHFT_LOG_PARM

```
CREATE TABLE M1_CREW_SHFT_LOG_PARM
  (
    CREW_SHFT_ID     CHAR(14 BYTE) NOT NULL ENABLE,
    SEQNO            NUMBER(5,0) NOT NULL ENABLE,
    PARM_SEQ         NUMBER(3,0) NOT NULL ENABLE,
    F1_MSG_PARM_VLONG VARCHAR2(2000 BYTE) DEFAULT '' NOT NULL ENABLE,
    MSG_PARM_TYP_FLG CHAR(4 BYTE),
    VERSION          NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
   CONSTRAINT M1_CREW_SHFT_LOG_PARM_FK FOREIGN KEY(CREW_SHFT_ID)
REFERENCES M1_CREW_SHFT
   )
PARTITION BY REFERENCE (M1_CREW_SHFT_LOG_PARM_FK)
ENABLE ROW MOVEMENT;


CREATE UNIQUE INDEX MT205P0 ON
M1_CREW_SHFT_LOG_PARM(CREW_SHFT_ID,SEQNO,PARM_SEQ) TABLESPACE <Index
Tablespace_Name>
GLOBAL PARTITION BY RANGE(CREW_SHFT_ID)
 (
 PARTITION "PART1" VALUES LESS THAN ('124999999999'),
 PARTITION "PART2" VALUES LESS THAN ('249999999999'),
 PARTITION "PART3" VALUES LESS THAN ('374999999999'),
 PARTITION "PART4" VALUES LESS THAN ('499999999999'),
 PARTITION "PART5" VALUES LESS THAN ('624999999999'),
 PARTITION "PART6" VALUES LESS THAN ('749999999999'),
 PARTITION "PART7" VALUES LESS THAN ('874999999999'),
 PARTITION "PART8" VALUES LESS THAN (MAXVALUE)
 );

ALTER TABLE M1_CREW_SHFT_LOG_PARM ADD CONSTRAINT MT205P0 PRIMARY
KEY(CREW_SHFT_ID,SEQNO,PARM_SEQ) USING INDEX;
```

M1_CREW_SHFT_MLOG

```
CREATE TABLE M1_CREW_SHFT_MLOG
  (
    CREW_SHFT_ID      CHAR(14 BYTE) NOT NULL ENABLE,
    SEQNO             NUMBER(5,0) NOT NULL ENABLE,
    BO_STATUS_REASON_CD VARCHAR2(30 BYTE),
    LOG_ENTRY_TYPE_FLG CHAR(4 BYTE) DEFAULT '' NOT NULL ENABLE,
    BO_STATUS_CD      CHAR(12 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_TYPE_CD      CHAR(8 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL          CHAR(16 BYTE) DEFAULT '' NOT NULL ENABLE,
    ADHOC_CHAR_VAL    VARCHAR2(254 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK1      VARCHAR2(50 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK2      VARCHAR2(50 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK3      VARCHAR2(50 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK4      VARCHAR2(50 BYTE) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK5      VARCHAR2(50 BYTE) DEFAULT '' NOT NULL ENABLE,
    DESCRLONG         VARCHAR2(4000 BYTE) DEFAULT '' NOT NULL ENABLE,
    MOB_LOG_DTTM DATE NOT NULL ENABLE,
    MESSAGE_CAT_NBR NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,
    MESSAGE_NBR     NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,
    USER_ID         CHAR(8 BYTE) DEFAULT '' NOT NULL ENABLE,
    VERSION         NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE ,
   CONSTRAINT M1_CREW_SHFT_MLOG_FK FOREIGN KEY(CREW_SHFT_ID)
REFERENCES M1_CREW_SHFT
   )
PARTITION BY REFERENCE (M1_CREW_SHFT_MLOG_FK)
```

```
ENABLE ROW MOVEMENT;

CREATE UNIQUE INDEX MT206P0 ON M1_CREW_SHFT_MLOG(CREW_SHFT_ID,SEQNO)
TABLESPACE <Index Tablespace_Name>
GLOBAL PARTITION BY RANGE(CREW_SHFT_ID)
 (
 PARTITION "PART1" VALUES LESS THAN ('124999999999'),
 PARTITION "PART2" VALUES LESS THAN ('249999999999'),
 PARTITION "PART3" VALUES LESS THAN ('374999999999'),
 PARTITION "PART4" VALUES LESS THAN ('499999999999'),
 PARTITION "PART5" VALUES LESS THAN ('624999999999'),
 PARTITION "PART6" VALUES LESS THAN ('749999999999'),
 PARTITION "PART7" VALUES LESS THAN ('874999999999'),
 PARTITION "PART8" VALUES LESS THAN (MAXVALUE)
 );

ALTER TABLE M1_CREW_SHFT_MLOG ADD CONSTRAINT MT206P0 PRIMARY
KEY(CREW_SHFT_ID,SEQNO) USING INDEX;
```

M1_CREW_SHFT_MLOG_PARM

```
CREATE TABLE M1_CREW_SHFT_MLOG_PARM
  (
    CREW_SHFT_ID      CHAR(14 BYTE) NOT NULL ENABLE,
    SEQNO             NUMBER(5,0) NOT NULL ENABLE,
    PARM_SEQ          NUMBER(3,0) NOT NULL ENABLE,
    F1_MSG_PARM_VLONG VARCHAR2(2000 BYTE) DEFAULT '' NOT NULL ENABLE,
    MSG_PARM_TYP_FLG  CHAR(4 BYTE),
    VERSION           NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE ,
  CONSTRAINT M1_CREW_SHFT_MLOG_PARM_FK FOREIGN KEY(CREW_SHFT_ID)
REFERENCES M1_CREW_SHFT
  )
PARTITION BY REFERENCE (M1_CREW_SHFT_MLOG_PARM_FK)
ENABLE ROW MOVEMENT;

CREATE UNIQUE INDEX MT207P0 ON
M1_CREW_SHFT_MLOG_PARM(CREW_SHFT_ID,SEQNO,PARM_SEQ) TABLESPACE <Index
Tablespace_Name>
GLOBAL PARTITION BY RANGE(CREW_SHFT_ID)
 (
 PARTITION "PART1" VALUES LESS THAN ('124999999999'),
 PARTITION "PART2" VALUES LESS THAN ('249999999999'),
 PARTITION "PART3" VALUES LESS THAN ('374999999999'),
 PARTITION "PART4" VALUES LESS THAN ('499999999999'),
 PARTITION "PART5" VALUES LESS THAN ('624999999999'),
 PARTITION "PART6" VALUES LESS THAN ('749999999999'),
 PARTITION "PART7" VALUES LESS THAN ('874999999999'),
 PARTITION "PART8" VALUES LESS THAN (MAXVALUE)
 );

ALTER TABLE M1_CREW_SHFT_MLOG_PARM ADD CONSTRAINT MT207P0 PRIMARY
KEY(CREW_SHFT_ID,SEQNO,PARM_SEQ) USING INDEX;
```

M1_CREW_SHFT_RESRC

```
CREATE TABLE M1_CREW_SHFT_RESRC
  (
    CREW_SHFT_ID CHAR(14 BYTE) NOT NULL ENABLE,
    SEQNO        NUMBER(5,0) NOT NULL ENABLE,
    RESRC_ID     CHAR(12 BYTE) NOT NULL ENABLE,
    EFF_DTTM DATE,
```

```
                    EXP_DTTM DATE,
                    BO_DATA_AREA CLOB,
                    VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE ,
                   CONSTRAINT M1_CREW_SHFT_RESRC_FK FOREIGN KEY(CREW_SHFT_ID)
REFERENCES M1_CREW_SHFT
                )
LOB (BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS
CACHE)
PARTITION BY REFERENCE (M1_CREW_SHFT_RESRC_FK)
ENABLE ROW MOVEMENT
;


CREATE UNIQUE INDEX MT200P0 ON M1_CREW_SHFT_RESRC(CREW_SHFT_ID,SEQNO)
TABLESPACE <Index Tablespace_Name>
GLOBAL PARTITION BY RANGE(CREW_SHFT_ID)
 (
 PARTITION "PART1" VALUES LESS THAN ('124999999999'),
 PARTITION "PART2" VALUES LESS THAN ('249999999999'),
 PARTITION "PART3" VALUES LESS THAN ('374999999999'),
 PARTITION "PART4" VALUES LESS THAN ('499999999999'),
 PARTITION "PART5" VALUES LESS THAN ('624999999999'),
 PARTITION "PART6" VALUES LESS THAN ('749999999999'),
 PARTITION "PART7" VALUES LESS THAN ('874999999999'),
 PARTITION "PART8" VALUES LESS THAN (MAXVALUE)
 );

ALTER TABLE M1_CREW_SHFT_RESRC ADD CONSTRAINT MT200P0 PRIMARY
KEY(CREW_SHFT_ID,SEQNO) USING INDEX;


CREATE INDEX MT200S1 ON M1_CREW_SHFT_RESRC(RESRC_ID)
GLOBAL PARTITION BY HASH(RESRC_ID)
 (
 PARTITION PART1 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART2 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART3 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART4 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART5 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART6 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART7 TABLESPACE <Index Tablespace_Name>,
 PARTITION PART8 TABLESPACE <Index Tablespace_Name>
);
```

M1_CREW_SHFT_SVC_AREA

```
CREATE TABLE M1_CREW_SHFT_SVC_AREA
  (
    CREW_SHFT_ID     CHAR(14 BYTE) NOT NULL ENABLE,
    SEQNO            NUMBER(5,0) NOT NULL ENABLE,
    SVC_AREA_CD      VARCHAR2(30 BYTE) NOT NULL ENABLE,
   SVC_AREA_USG_FLG CHAR(4 BYTE) DEFAULT '' NOT NULL ENABLE,
    VERSION          NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE ,
    CONSTRAINT M1_CREW_SHFT_SVC_AREA_FK FOREIGN KEY(CREW_SHFT_ID)
REFERENCES M1_CREW_SHFT
    )
PARTITION BY REFERENCE (M1_CREW_SHFT_SVC_AREA_FK)
ENABLE ROW MOVEMENT;
```

```
                  CREATE UNIQUE INDEX MT202P0 ON
                  M1_CREW_SHFT_SVC_AREA(CREW_SHFT_ID,SEQNO) TABLESPACE <Index
                  Tablespace_Name>
                  GLOBAL PARTITION BY RANGE(CREW_SHFT_ID)
                   (
                   PARTITION "PART1" VALUES LESS THAN ('124999999999'),
                   PARTITION "PART2" VALUES LESS THAN ('249999999999'),
                   PARTITION "PART3" VALUES LESS THAN ('374999999999'),
                   PARTITION "PART4" VALUES LESS THAN ('499999999999'),
                   PARTITION "PART5" VALUES LESS THAN ('624999999999'),
                   PARTITION "PART6" VALUES LESS THAN ('749999999999'),
                   PARTITION "PART7" VALUES LESS THAN ('874999999999'),
                   PARTITION "PART8" VALUES LESS THAN (MAXVALUE)
                   );

                  ALTER TABLE M1_CREW_SHFT_SVC_AREA ADD CONSTRAINT MT202P0 PRIMARY
                  KEY(CREW_SHFT_ID,SEQNO) USING INDEX;


                  CREATE UNIQUE INDEX MT202S2 ON
                  M1_CREW_SHFT_SVC_AREA(CREW_SHFT_ID,SVC_AREA_CD,SEQNO)
                  GLOBAL PARTITION BY HASH(CREW_SHFT_ID)
                   (
                   PARTITION PART1 TABLESPACE <Index Tablespace_Name>,
                   PARTITION PART2 TABLESPACE <Index Tablespace_Name>,
                   PARTITION PART3 TABLESPACE <Index Tablespace_Name>,
                   PARTITION PART4 TABLESPACE <Index Tablespace_Name>,
                   PARTITION PART5 TABLESPACE <Index Tablespace_Name>,
                   PARTITION PART6 TABLESPACE <Index Tablespace_Name>,
                   PARTITION PART7 TABLESPACE <Index Tablespace_Name>,
                   PARTITION PART8 TABLESPACE <Index Tablespace_Name>
                  );
```

M1_CREW_SHFT_SVC_CLS

```
                  CREATE TABLE M1_CREW_SHFT_SVC_CLS
                    (
                      CREW_SHFT_ID    CHAR(14 BYTE) NOT NULL ENABLE,
                      SEQNO           NUMBER(5,0) NOT NULL ENABLE,
                      SVC_CLS_CD      VARCHAR2(30 BYTE) DEFAULT '' NOT NULL ENABLE,
                      SVC_CLS_USG_FLG CHAR(4 BYTE) DEFAULT '' NOT NULL ENABLE,
                      VERSION         NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE ,
                     CONSTRAINT M1_CREW_SHFT_SVC_CLS_FK FOREIGN KEY(CREW_SHFT_ID)
                  REFERENCES M1_CREW_SHFT
                     )
                  PARTITION BY REFERENCE (M1_CREW_SHFT_SVC_CLS_FK)
                  ENABLE ROW MOVEMENT;


                  CREATE UNIQUE INDEX MT203P0 ON
                  M1_CREW_SHFT_SVC_CLS(CREW_SHFT_ID,SEQNO) TABLESPACE <Index
                  Tablespace_Name>
                  GLOBAL PARTITION BY RANGE(CREW_SHFT_ID)
                   (
                   PARTITION "PART1" VALUES LESS THAN ('124999999999'),
                   PARTITION "PART2" VALUES LESS THAN ('249999999999'),
                   PARTITION "PART3" VALUES LESS THAN ('374999999999'),
                   PARTITION "PART4" VALUES LESS THAN ('499999999999'),
                   PARTITION "PART5" VALUES LESS THAN ('624999999999'),
                   PARTITION "PART6" VALUES LESS THAN ('749999999999'),
                   PARTITION "PART7" VALUES LESS THAN ('874999999999'),
                   PARTITION "PART8" VALUES LESS THAN (MAXVALUE)
```

```
 );

ALTER TABLE M1_CREW_SHFT_SVC_CLS ADD CONSTRAINT MT203P0 PRIMARY
KEY(CREW_SHFT_ID,SEQNO) USING INDEX;

CREATE UNIQUE INDEX MT203S1 ON
M1_CREW_SHFT_SVC_CLS(SVC_CLS_CD,CREW_SHFT_ID,SEQNO,SVC_CLS_USG_FLG) ;
```

Appendix E

Sample Statements and DDL for ILM Enabled Maintenance Objects

This section provides additional details including the sample syntax for each step using the To Do Entry maintenance object as an example. Other maintenance object's implementations can follow a similar pattern.

1. Rename existing table CI_TD_ENTRY and primary key index as a backup. It is suggested to use an ILM_ prefix. The following are sample statements:

   ```
   ALTER TABLE CI_TD_ENTRY RENAME TO ILM_TD_ENTRY;
   ALTER INDEX XT039P0 RENAME TO ILM_XT039P0;
   ```

2. Generate DDL for the secondary index.

   ```
   set heading off;
   set echo off;
   Set pages 999;
   set long 90000;

   spool ddl_list.sql
   select dbms_metadata.get_ddl('INDEX','XT039S2','CISADM') from dual;
   select dbms_metadata.get_ddl('INDEX','XT039S3','CISADM') from dual;
   select dbms_metadata.get_ddl('INDEX','XT039S4','CISADM') from dual;
   select dbms_metadata.get_ddl('INDEX','XT039S5','CISADM') from dual;
   select dbms_metadata.get_ddl('INDEX','XT039S6','CISADM') from dual;
   select dbms_metadata.get_ddl('INDEX','XT039S7','CISADM') from dual;
   select dbms_metadata.get_ddl('INDEX','XT039S8','CISADM') from dual;
   spool off;
   ```

3. Drop secondary indexes.

   ```
   DROP INDEX CISADM.XT039S2;
   DROP INDEX CISADM.XT039S3;
   DROP INDEX CISADM.XT039S4;
   DROP INDEX CISADM.XT039S5;
   DROP INDEX CISADM.XT039S6;
   DROP INDEX CISADM.XT039S7;
   DROP INDEX CISADM.XT039S8;
   ```

4. Create Partitioned Table.

   In the following example ILM_DT value is inserted from column CRE_DTTM. The degree setting of 'parallel' in the DDL can be adjusted according to the table's data, its means and its size.

```
CREATE TABLE CI_TD_ENTRY  nologging  parallel (degree 10)
PARTITION BY RANGE (ILM_DT) SUBPARTITION BY RANGE (TD_ENTRY_ID)
SUBPARTITION TEMPLATE
(
SUBPARTITION SUB1 VALUES LESS THAN ( '124999999999' ),
SUBPARTITION SUB2 VALUES LESS THAN ( '249999999999' ),
SUBPARTITION SUB3 VALUES LESS THAN ( '374999999999' ),
SUBPARTITION SUB4 VALUES LESS THAN ( '499999999999' ),
SUBPARTITION SUB5 VALUES LESS THAN ( '624999999999' ),
SUBPARTITION SUB6 VALUES LESS THAN ( '749999999999' ),
SUBPARTITION SUB7 VALUES LESS THAN ( '874999999999' ),
SUBPARTITION SUB8 VALUES LESS THAN ( MAXVALUE )
)
(
PARTITION P1 VALUES LESS THAN(TO_DATE('2012-01-01 00:00:00','SYYYY-
MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIAN')),
PARTITION P2 VALUES LESS THAN(TO_DATE('2013-01-01 00:00:00','SYYYY-
MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIAN')),
PARTITION P3 VALUES LESS THAN(TO_DATE('2014-01-01 00:00:00','SYYYY-
MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIAN')),
PARTITION P4 VALUES LESS THAN(TO_DATE('2015-01-01 00:00:00','SYYYY-
MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIAN')),
PARTITION P5 VALUES LESS THAN(TO_DATE('2015-04-01 00:00:00','SYYYY-
MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIAN')),
PARTITION P6 VALUES LESS THAN(TO_DATE('2015-07-01 00:00:00','SYYYY-
MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIAN')),
PARTITION P7 VALUES LESS THAN(TO_DATE('2015-10-01 00:00:00','SYYYY-
MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIAN')),
PARTITION P8 VALUES LESS THAN(TO_DATE('2016-01-01 00:00:00','SYYYY-
MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIAN')),
PARTITION P9 VALUES LESS THAN (MAXVALUE)
)ENABLE ROW MOVEMENT
as select
TD_ENTRY_ID,
BATCH_CD,
BATCH_NBR,
MESSAGE_CAT_NBR,
MESSAGE_NBR,
ASSIGNED_TO,
TD_TYPE_CD,
ROLE_ID,
ENTRY_STATUS_FLG,
VERSION,
CRE_DTTM,
ASSIGNED_DTTM,
COMPLETE_DTTM,
COMPLETE_USER_ID,
COMMENTS,
ASSIGNED_USER_ID,
TD_PRIORITY_FLG,
CRE_DTTM as ILM_DT,
ILM_ARCH_SW
from ILM_TD_ENTRY
/
```

5. Enable logging option for table CI_TD_ENRY.

```
ALTER TABLE CI_TD_ENTRY LOGGING;
```

6. Create Primary Index for Parent table CI_TD_ENTRY.

```
CREATE UNIQUE INDEX XT039P0 ON CI_TD_ENTRY NOLOGGING (
TD_ENTRY_ID
```

```
    )
    GLOBAL PARTITION BY RANGE (TD_ENTRY_ID) (
    PARTITION P1 VALUES LESS THAN ( '124999999999' ), PARTITION P2
VALUES LESS THAN ( '249999999999' ), PARTITION P3 VALUES LESS THAN (
'374999999999' ), PARTITION P4 VALUES LESS THAN ( '499999999999' ),
PARTITION P5 VALUES LESS THAN ( '624999999999' ), PARTITION P6 VALUES
LESS THAN ( '744999999999' ), PARTITION P7 VALUES LESS THAN (
'874999999999' ), PARTITION P8 VALUES LESS THAN ( MAXVALUE )
    )
    /
```

7. Add Primary Key for Parent table CI_TD_ENTRY

```
    ALTER TABLE CI_TD_ENTRY ADD CONSTRAINT XT039P0 PRIMARY
KEY(TD_ENTRY_ID) USING INDEX
    /
```

8. Create Secondary Indexes for Parent table CI_TD_ENTRY

```
    CREATE UNIQUE INDEX ILM_XT039S1 ON CI_TD_ENTRY NOLOGGING (ILM_DT,
ILM_ARCH_SW, TD_ENTRY_ID)LOCAL(
    PARTITION I_P1,
    PARTITION I_P2,
    PARTITION I_P3,
    PARTITION I_P4,
    PARTITION I_P5,
    PARTITION I_P6,
    PARTITION I_P7,
    PARTITION I_P8,
    PARTITION I_P9
    )

    CREATE UNIQUE INDEX XT039S2 ON CI_TD_ENTRY (
    ASSIGNED_TO, TD_ENTRY_ID
    )
    /

    CREATE INDEX XT039S3 ON CI_TD_ENTRY (
    ENTRY_STATUS_FLG, ASSIGNED_TO
    )
    /



    CREATE INDEX XT039S4 ON CI_TD_ENTRY (
    ROLE_ID, TD_TYPE_CD, ENTRY_STATUS_FLG, TD_PRIORITY_FLG
    )
    /

    CREATE INDEX XT039S5 ON CI_TD_ENTRY (
    BATCH_CD, BATCH_NBR, ENTRY_STATUS_FLG
    )
    /

    CREATE UNIQUE INDEX XT039S6 ON CI_TD_ENTRY (
    TD_ENTRY_ID, ASSIGNED_TO, ENTRY_STATUS_FLG
    )
    /
    CREATE UNIQUE INDEX XT039S7 ON CI_TD_ENTRY (
    COMPLETE_USER_ID, COMPLETE_DTTM, TD_ENTRY_ID
    )
```

9. After verification of the ILM based tables, user can drop the backup tables "ILM" renamed table.

10. Create Child Tables, Primary Key, Primary Indexes and Secondary Indexes as shown below.

Create Child Table CI_TD_ENTRY_CHA

```
CREATE TABLE CI_TD_ENTRY_CHA (
TD_ENTRY_ID   CHAR(14 BYTE) NOT NULL ENABLE, CHAR_TYPE_CD   CHAR(8
BYTE) NOT NULL ENABLE,
SEQ_NUM        NUMBER(3,0) DEFAULT 0 NOT NULL ENABLE, CHAR_VAL
CHAR(16 BYTE) DEFAULT ' ' NOT NULL ENABLE, VERSION        NUMBER(5,0)
DEFAULT 1 NOT NULL ENABLE, ADHOC_CHAR_VAL VARCHAR2(254 BYTE) DEFAULT '
' NOT NULL ENABLE, CHAR_VAL_FK1   VARCHAR2(50 BYTE) DEFAULT ' ' NOT
NULL ENABLE, CHAR_VAL_FK2   VARCHAR2(50 BYTE) DEFAULT ' ' NOT NULL
ENABLE, CHAR_VAL_FK3   VARCHAR2(50 BYTE) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK4   VARCHAR2(50 BYTE) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK5   VARCHAR2(50 BYTE) DEFAULT ' ' NOT NULL ENABLE,
SRCH_CHAR_VAL  VARCHAR2(50 BYTE) DEFAULT ' ' NOT NULL ENABLE,
CONSTRAINT CI_TD_ENTRY_CHA_FK FOREIGN KEY(TD_ENTRY_ID) REFERENCES
CI_TD_ENTRY)
PARTITION BY REFERENCE (CI_TD_ENTRY_CHA_FK) ENABLE ROW MOVEMENT
/
```

Create Primary Index for Child Table CI_TD_ENTRY_CHA

```
CREATE UNIQUE INDEX XT701P0 ON CI_TD_ENTRY_CHA (
TD_ENTRY_ID, CHAR_TYPE_CD, SEQ_NUM
)
GLOBAL PARTITION BY RANGE (TD_ENTRY_ID) (
PARTITION P1 VALUES LESS THAN ( '124999999999' ), PARTITION P2
VALUES LESS THAN ( '249999999999' ), PARTITION P3 VALUES LESS THAN (
'374999999999' ), PARTITION P4 VALUES LESS THAN ( '499999999999' ),
PARTITION P5 VALUES LESS THAN ( '624999999999' ), PARTITION P6 VALUES
LESS THAN ( '744999999999' ), PARTITION P7 VALUES LESS THAN (
'874999999999' ), PARTITION P8 VALUES LESS THAN ( MAXVALUE )
)
/
```

Create Primary Key for Child Table CI_TD_ENTRY_CHA

```
ALTER TABLE CI_TD_ENTRY_CHA ADD CONSTRAINT XT701P0 PRIMARY
KEY(TD_ENTRY_ID, CHAR_TYPE_CD, SEQ_NUM) USING INDEX
/
```

Create Secondary Indexes for Child Table CI_TD_ENTRY_CHA

```
CREATE INDEX XT701S1 ON CI_TD_ENTRY_CHA (
SRCH_CHAR_VAL, CHAR_TYPE_CD, TD_ENTRY_ID
)
/
```

Create Child Table CI_TD_DRLKEY

```
CREATE TABLE CI_TD_DRLKEY (
TD_ENTRY_ID CHAR(14 BYTE) NOT NULL ENABLE, SEQ_NUM    NUMBER(3,0)
NOT NULL ENABLE,
KEY_VALUE  VARCHAR2(50 BYTE) DEFAULT ' ' NOT NULL ENABLE, VERSION
NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
CONSTRAINT CI_TD_DRLKEY_FK FOREIGN KEY(TD_ENTRY_ID) REFERENCES
CI_TD_ENTRY) PARTITION BY REFERENCE (CI_TD_DRLKEY_FK)
```

```
ENABLE ROW MOVEMENT
/
```

Create Primary Index for Child Table CI_TD_DRLKEY

```
CREATE UNIQUE INDEX XT037P0 ON CI_TD_DRLKEY (
TD_ENTRY_ID, SEQ_NUM
)
GLOBAL PARTITION BY RANGE (TD_ENTRY_ID) (
PARTITION P1 VALUES LESS THAN ( '124999999999' ), PARTITION P2
VALUES LESS THAN ( '249999999999' ), PARTITION P3 VALUES LESS THAN (
'374999999999' ), PARTITION P4 VALUES LESS THAN ( '499999999999' ),
PARTITION P5 VALUES LESS THAN ( '624999999999' ), PARTITION P6 VALUES
LESS THAN ( '744999999999' ), PARTITION P7 VALUES LESS THAN (
'874999999999' ), PARTITION P8 VALUES LESS THAN ( MAXVALUE )
)
/
```

Create Primary Key for Child Table CI_TD_DRLKEY

```
ALTER TABLE CI_TD_DRLKEY ADD CONSTRAINT XT037P0 PRIMARY
KEY(TD_ENTRY_ID, SEQ_NUM) USING INDEX
/
```

Create Secondary Indexes for Child Table CI_TD_DRLKEY

```
CREATE INDEX XT037S1 ON CI_TD_DRLKEY (
KEY_VALUE, TD_ENTRY_ID
)
/
```

Create Child Table CI_TD_LOG

```
CREATE TABLE CI_TD_LOG (
TD_ENTRY_ID CHAR(14 BYTE) NOT NULL ENABLE, SEQ_NUM    NUMBER(3,0)
NOT NULL ENABLE, LOG_DTTM DATE NOT NULL ENABLE,
LOG_TYPE_FLG CHAR(4 BYTE) DEFAULT ' ' NOT NULL ENABLE, USER_ID
CHAR(8 BYTE) DEFAULT ' ' NOT NULL ENABLE, ASSIGNED_TO CHAR(8 BYTE)
DEFAULT ' ' NOT NULL ENABLE, VERSION    NUMBER(5,0) DEFAULT 1 NOT NULL
ENABLE,
DESCRLONG   VARCHAR2(4000 BYTE) DEFAULT ' ' NOT NULL ENABLE,
CONSTRAINT CI_TD_LOG_FK FOREIGN KEY(TD_ENTRY_ID) REFERENCES
CI_TD_ENTRY) PARTITION BY REFERENCE (CI_TD_LOG_FK)
ENABLE ROW MOVEMENT
/
```

Create Unique Index for Child Table CI_TD_LOG

```
CREATE UNIQUE INDEX XT721P0 ON CI_TD_LOG (
TD_ENTRY_ID, SEQ_NUM
)
GLOBAL PARTITION BY RANGE (TD_ENTRY_ID) (
PARTITION P1 VALUES LESS THAN ( '124999999999' ), PARTITION P2
VALUES LESS THAN ( '249999999999' ), PARTITION P3 VALUES LESS THAN (
'374999999999' ), PARTITION P4 VALUES LESS THAN ( '499999999999' ),
PARTITION P5 VALUES LESS THAN ( '624999999999' ), PARTITION P6 VALUES
LESS THAN ( '744999999999' ), PARTITION P7 VALUES LESS THAN (
'874999999999' ), PARTITION P8 VALUES LESS THAN ( MAXVALUE )
)
/
```

Create Primary Key for Child Table CI_TD_LOG

```
ALTER TABLE CI_TD_LOG ADD CONSTRAINT XT721P0 PRIMARY
KEY(TD_ENTRY_ID, SEQ_NUM) USING INDEX
    /
```

Create Index on Child Table CI_TD_LOG

```
CREATE INDEX XT721S1 ON CI_TD_LOG (
LOG_DTTM,
USER_ID, LOG_TYPE_FLG, TD_ENTRY_ID
)
    /
```

Create Child Table CI_TD_MSG_PRM

```
CREATE TABLE CI_TD_MSG_PARM (
TD_ENTRY_ID CHAR(14 BYTE) NOT NULL ENABLE, SEQ_NUM     NUMBER(3,0)
NOT NULL ENABLE,
    MSG_PARM_VAL VARCHAR2(30 BYTE) DEFAULT ' ' NOT NULL ENABLE, VERSION
NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    CONSTRAINT CI_TD_MSG_PARM_FK FOREIGN KEY(TD_ENTRY_ID) REFERENCES
CI_TD_ENTRY)
    PARTITION BY REFERENCE (CI_TD_MSG_PARM_FK) ENABLE ROW MOVEMENT
    /
```

Create Unique Index for Child Table CI_TD_MSG_PRM
```
CREATE UNIQUE INDEX XT040P0 ON CI_TD_MSG_PARM (
TD_ENTRY_ID, SEQ_NUM
)
    /
```

Create Primary Key for Child Table CI_TD_MSG
```
ALTER TABLE CI_TD_MSG_PARM ADD CONSTRAINT XT040P0 PRIMARY
KEY(TD_ENTRY_ID, SEQ_NUM) USING INDEX
    /
```

Create Child Table CI_TD_SRTKEY
```
CREATE TABLE CI_TD_SRTKEY (
TD_ENTRY_ID CHAR(14 BYTE) NOT NULL ENABLE, SEQ_NUM     NUMBER(3,0)
NOT NULL ENABLE,
    KEY_VALUE  VARCHAR2(50 BYTE) DEFAULT ' ' NOT NULL ENABLE, VERSION
NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    CONSTRAINT CI_TD_SRTKEY_FK FOREIGN KEY(TD_ENTRY_ID) REFERENCES
CI_TD_ENTRY) PARTITION BY REFERENCE (CI_TD_SRTKEY_FK)
    ENABLE ROW MOVEMENT
    /
```

Create Unique Index for Child Table CI_TD_SRTKEY
```
CREATE UNIQUE INDEX XT041P0 ON CI_TD_SRTKEY (
TD_ENTRY_ID, SEQ_NUM
)
    GLOBAL PARTITION BY RANGE (TD_ENTRY_ID) (
    PARTITION P1 VALUES LESS THAN ( '124999999999' ), PARTITION P2
VALUES LESS THAN ( '249999999999' ), PARTITION P3 VALUES LESS THAN (
'374999999999' ), PARTITION P4 VALUES LESS THAN ( '499999999999' ),
PARTITION P5 VALUES LESS THAN ( '624999999999' ), PARTITION P6 VALUES
LESS THAN ( '744999999999' ), PARTITION P7 VALUES LESS THAN (
'874999999999' ), PARTITION P8 VALUES LESS THAN ( MAXVALUE )
    )
    /
```

Create Primary Key for Child Table CI_TD_STRKEY

```
    ALTER TABLE CI_TD_SRTKEY ADD CONSTRAINT XT041P0 PRIMARY
KEY(TD_ENTRY_ID, SEQ_NUM) USING INDEX
    /
```

Create Index for Child Table CI_TD_STRKEY

```
    CREATE INDEX XT041S1 ON CI_TD_SRTKEY (
    KEY_VALUE, TD_ENTRY_ID
    )
    /
```

Create Child Table CI_TD_ENTRY_K

```
    CREATE TABLE CI_TD_ENTRY_K (
    TD_ENTRY_ID CHAR(14 BYTE) NOT NULL ENABLE, ENV_ID     NUMBER(6,0)
NOT NULL ENABLE,
    CONSTRAINT XT547P0 PRIMARY KEY (TD_ENTRY_ID, ENV_ID) ENABLE
    )
    /
```