Oracle Utilities Customer To Meter
Database Administrator’s Guide
Release 2.6.0.1.0
E92494-01

February 2018
# Contents

Preface........................................................................................................................................................................... i-i

Audience ........................................................................................................................................................................... i-i

Related Documents ......................................................................................................................................................... i-i

Conventions..................................................................................................................................................................... i-ii

Acronyms......................................................................................................................................................................... i-ii

## Chapter 1

Database Overview .............................................................................................................................................................................. 1-1

Supported Platforms Summary Table ................................................................................................................................. 1-2

Support for Software Patches and Upgrades ......................................................................................................................... 1-2

Database Maintenance Rules .................................................................................................................................................. 1-3

Permitted Database Changes .............................................................................................................................................. 1-3

Non-Permitted Database Changes ........................................................................................................................................ 1-3

## Chapter 2

Installing the Version 2.6.0.1.0 Database ........................................................................................................................................... 2-1

Creating the Database ................................................................................................................................................................. 2-2

Oracle Database Installation ......................................................................................................................................................... 2-4

Database Scripts and Utilities ...................................................................................................................................................... 2-4

Initial Install or Installing Version 2.6.0.1.0 for the First Time ................................................................................................. 2-4

Upgrade Install ......................................................................................................................................................................... 2-26

Demo Install ............................................................................................................................................................................. 2-28

## Chapter 3

Database Design .............................................................................................................................................................................. 3-1

Categories of Data ........................................................................................................................................................................ 3-2

Naming Standards ....................................................................................................................................................................... 3-2

Column Data Type and Constraints ........................................................................................................................................... 3-5

User Defined Code ...................................................................................................................................................................... 3-6

System Assigned Identifier ......................................................................................................................................................... 3-6

Date/Time/Timestamp .................................................................................................................................................................. 3-6

Number ....................................................................................................................................................................................... 3-6

Fixed Length/Variable Length Character Columns .................................................................................................................. 3-6

Null Column Support ................................................................................................................................................................. 3-6

XML Type Support .................................................................................................................................................................... 3-7

Cache and Key Validation Flags .............................................................................................................................................. 3-7

Table Classification and Table Volume Flags ......................................................................................................................... 3-7

Default Value Setting ............................................................................................................................................................... 3-7

Foreign Key Constraints ............................................................................................................................................................ 3-8

Standard Columns ....................................................................................................................................................................... 3-8

Owner Flag ............................................................................................................................................................................... 3-8

Version ..................................................................................................................................................................................... 3-8

## Chapter 4

Database Implementation Guidelines ............................................................................................................................................... 4-1
Chapter 5
Conversion Tools ............................................................................................................................................................... 5-1
  Script Installation ....................................................................................................................................................... 5-2
  Preparing the Production Database ....................................................................................................................... 5-3
  Preparing the Staging Database ............................................................................................................................... 5-3

Chapter 6
Information Lifecycle Management and CC&B Data Archiving in C2M ................................................................. 6-1
  ILM Implementation Overview ............................................................................................................................... 6-2
  ILM Implementation Components ........................................................................................................................... 6-2
  ILM Database Administrator’s Tasks ........................................................................................................................ 6-3
    Preparation Phase ................................................................................................................................................ 6-3
    On-going Maintenance Phase .............................................................................................................................. 6-4
  Naming Convention .............................................................................................................................................. 6-6

Chapter 7
Information Lifecycle Management and MDM Data Archiving in C2M ................................................................. 7-1
  ILM Implementation Overview ............................................................................................................................... 7-2
  ILM Implementation Components ........................................................................................................................... 7-2
  ILM Database Administrator’s Tasks ........................................................................................................................ 7-3
    Preparation Phase ................................................................................................................................................ 7-3
    On-going Maintenance Phase .............................................................................................................................. 7-40
  Naming Convention .............................................................................................................................................. 7-42

Appendix A
Sample SQL for Enabling ILM in C2M for CC&B (Initial Install) ............................................................................... A-1
  Maintenance Object: TO DO ENTRY ....................................................................................................................... A-1
    Parent Table: CI_TD_ENTRY ................................................................................................................................ A-1
    Child Table: CI_TD_DRLKEY ............................................................................................................................... A-4
    Child Table: CI_TD_ENTRY_CHA ........................................................................................................................ A-5
    Child Table: CI_TD_LOG ................................................................................................................................... A-5
    Child Table: CI_TD_MSG_PARM ........................................................................................................................ A-6
    Child Table: CI_TD_SRTKEY ............................................................................................................................... A-7
  Maintenance Object:F1-SYNCREQIN .................................................................................................................... A-7
    Parent Table: F1_SYNC_REQ_IN ........................................................................................................................ A-7
    Child Table: F1_SYNC_REQ_IN_CHAR ................................................................................................................ A-12
    Child Table: F1_SYNC_REQ_IN_EXCP .............................................................................................................. A-13
    Child Table: F1_SYNC_REQ_IN_EXCP_PARM ............................................................................................... A-13
    Child Table: F1_SYNC_REQ_IN_LOG ................................................................................................................ A-14
    Child Table: F1_SYNC_REQ_IN_LOG_PARM .................................................................................................... A-15
    Child Table: F1_SYNC_REQ_IN_REL_OBJ ...................................................................................................... A-15

Appendix B
Sample SQL For Enabling ILM in C2M for CC&B (Existing Installation) ................................................................. B-1

Appendix C
Sample SQL for Periodic Maintenance for CC&B Data ........................................................................................................ C-1
  Archive Partition .................................................................................................................................................................. C-2
  Restore Partition .................................................................................................................................................................. C-5

Appendix D
Sample SQL for Partitioning with ILM in C2M for CC&B .................................................................................................. D-1
  Parent Table: CI_ADJ ......................................................................................................................................................... D-2
  Parent Table: CI_BSEG ..................................................................................................................................................... D-9

Appendix E
Sample SQL for Enabling ILM in C2M for MDM (Initial Install) ..................................................................................... E-1
  Maintenance Object: TO DO ENTRY ................................................................................................................................. E-1
  Parent Table: CL_TD_ENTRY ............................................................................................................................................... E-1
  Child Table: CI_TD_DRLKEY .............................................................................................................................................. E-4
  Child Table: CI_TDENTRY_CHA ......................................................................................................................................... E-5
  Child Table: CI_TD_LOG ...................................................................................................................................................... E-5
  Child Table: CI_TD_MSG_PARM ....................................................................................................................................... E-6
  Child Table: CI_TD_SRTKEY .............................................................................................................................................. E-7
  Parent Table: F1_SYNC_REQ_IN ....................................................................................................................................... E-7
  Child Table: F1_SYNC_REQ_IN_CHAR ................................................................................................................................. E-12
  Child Table: F1_SYNC_REQ_IN_EXCP ................................................................................................................................. E-12
  Child Table: F1_SYNC_REQ_IN_EXCP_PARM ...................................................................................................................... E-13
  Child Table: F1_SYNC_REQ_IN_LOG ................................................................................................................................ E-14
  Child Table: F1_SYNC_REQ_IN_LOG_PARM ......................................................................................................................... E-15
  Child Table: F1_SYNC_REQ_IN_REL_OBJ .......................................................................................................................... E-15
  Parent Table: D1_INIT_MSRMT_DATA ................................................................................................................................. E-16
  Child Table: D1_INIT_MSRMT_DATA_CHAR ......................................................................................................................... E-25
  Child Table: D1_INIT_MSRMT_DATA_LOG ......................................................................................................................... E-26
  Child Table: D1_INIT_MSRMT_DATA_LOG_PARM ................................................................................................................ E-27
  Child Table: D1_INIT_MSRMT_DATA_K .............................................................................................................................. E-27

Appendix F
Sample SQL For Enabling ILM in C2M for MDM (Existing Installation) ........................................................................... F-1

Appendix G
Sample SQL for ILM in C2M with Sub Retention (Existing Installation) ............................................................................... G-1

Appendix H
Sample SQL for Periodic Maintenance for MDM Data ........................................................................................................ H-1
  Adding Partition ................................................................................................................................................................. H-2
  Archiving Partition ........................................................................................................................................................... H-2
  Archiving Subpartition ....................................................................................................................................................... H-5
  Restoring Partition ............................................................................................................................................................ H-7
  Restoring Subpartition ...................................................................................................................................................... H-8
  Compressing Partition (D1_MSRMT table only) ................................................................................................................ H-9

Appendix I
Sample Scripts for Customer Contact Enhancement ........................................................................................................... I-1
  Updating Preferred Contact Method on Legacy Values ................................................................................................... I-7

Appendix J
Partitioning and Compression Recommendations ................................................................................................................. J-1
  Partitioning Recommendations ......................................................................................................................................... J-2
    D1_MSRMT ..................................................................................................................................................................... J-2
    D1_MSRMT_CHAR ......................................................................................................................................................... J-3
    D1_MSRMT_LOG .......................................................................................................................................................... J-4
    D1_MSRMT_LOG_PARM ................................................................................................................................................ J-6
Appendix K
Upgrades to the Oracle Utilities Customer To Meter 2.6.0.1.0 Database ........................................ K-1
  Schema Changes ................................................................. K-2
  New System Data .............................................................. K-2
  New Tables ................................................................. K-24
  New Views ................................................................. K-24
  Dropped Columns .......................................................... K-24
  Added Columns ............................................................. K-24
  Renamed Columns .......................................................... K-24
  Column Format Change .................................................. K-25
  Primary Key Change ........................................................ K-25
  Added Indexes .............................................................. K-25
  Dropped Indexes ........................................................... K-26
  Index Changes .............................................................. K-26

Appendix L
Upgrades to the Oracle Utilities Application Framework Database ................................................. L-1
  Upgrading from Oracle Utilities Application Framework v4.3.0.4.0 to v4.3.0.5.0 ........................................ L-2

Appendix M
Oracle Utilities Customer To Meter System Table Guide .......................................................... M-1
  Installation Options ........................................................ M-2

Appendix N
Oracle Application Framework System Table Guide .............................................................. N-1
  System Table Standards ..................................................... N-2
  Business Configuration Tables .......................................... N-3
  Development and Implementation System Tables ......................... N-5
Preface

This guide provides instructions for installing and maintaining the database for Oracle Utilities Customer To Meter.

Audience

Database Administrator’s Guide is intended for database administrators who will be installing and maintaining the database for Oracle Utilities Customer To Meter.

Related Documents

For more information, refer to these Oracle documents:

Installation Guides and Release Notes

- Oracle Utilities Customer To Meter V2.6.0.1.0 Release Notes
- Oracle Utilities Customer To Meter V2.6.0.1.0 Quick Install Guide
- Oracle Utilities Customer To Meter V2.6.0.1.0 Installation Guide
- Oracle Utilities Customer To Meter V2.6.0.1.0 Database Administrator’s Guide
- Oracle Utilities Customer To Meter V2.6.0.1.0 Optional Products Installation Guide
- Oracle Utilities Customer To Meter V2.6.0.1.0 Licensing Information User Manual

Administrative and Business User Guides

- Oracle Utilities Customer To Meter V2.6.0.1.0 Administrative User Guide
- Oracle Utilities Customer To Meter V2.6.0.1.0 Business User Guide

Supplemental Documents

- Oracle Utilities Customer To Meter V2.6.0.1.0 Server Administration Guide
- Oracle Utilities Customer To Meter V2.6.0.1.0 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:

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

Acronyms

The following terms are used in this document:

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPEL</td>
<td>Business Process Execution Language</td>
</tr>
<tr>
<td>C2M</td>
<td>Customer To Meter</td>
</tr>
<tr>
<td>CCB (or CC&amp;B)</td>
<td>Customer Care and Billing</td>
</tr>
<tr>
<td>MDM</td>
<td>Meter Data Management</td>
</tr>
<tr>
<td>OUAF</td>
<td>Oracle Utilities Application Framework</td>
</tr>
<tr>
<td>OSB</td>
<td>Oracle Service Bus</td>
</tr>
<tr>
<td>SOA</td>
<td>Service Oriented Architecture</td>
</tr>
<tr>
<td>SOM</td>
<td>Service Order Management</td>
</tr>
<tr>
<td>SGG</td>
<td>Smart Grid Gateway</td>
</tr>
</tbody>
</table>
Chapter 1
Database Overview

This section provides an overview of the Oracle Utilities Customer To Meter database, including:

- Supported Database Platforms
- Database Maintenance Rules
Supported Database Platforms

This section defines the platforms on which Oracle Utilities Customer To Meter is verified to operate.

Supported Platforms Summary Table

Oracle Utilities Customer To Meter is supported on the following platforms:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Database Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX 7.2 TL4 (POWER 64-bit)</td>
<td>Oracle DB 12.1.0.2+ (64-bit)</td>
</tr>
<tr>
<td></td>
<td>Oracle DB 12.2.0.1+ (64-bit)</td>
</tr>
<tr>
<td>Linux 7.1 (64-bit) x86_64 (64-bit)</td>
<td>Oracle DB 12.1.0.2+ (64-bit)</td>
</tr>
<tr>
<td></td>
<td>Oracle DB 12.2.0.1+ (64-bit)</td>
</tr>
<tr>
<td>Solaris 11 (SPARC 64-bit)</td>
<td>Oracle DB 12.1.0.2+ (64-bit)</td>
</tr>
<tr>
<td></td>
<td>Oracle DB 12.2.0.1+ (64-bit)</td>
</tr>
<tr>
<td>HP UX 11.31 (64-bit)</td>
<td>Oracle DB 12.1.0.2+ (64-bit)</td>
</tr>
<tr>
<td></td>
<td>Oracle DB 12.2.0.1+ (64-bit)</td>
</tr>
</tbody>
</table>

* Oracle Utilities Customer To Meter is tested on both Oracle Database Enterprise Edition and Standard Edition. Some features, such as Advanced Compression and Partitioning, require the Enterprise Edition.

The following Oracle Database Server Editions are supported:

- Oracle Database Enterprise Edition
- Oracle Database Standard Edition

Note: Oracle Database Enterprise Edition and the Partitioning and Advanced Compression options are not mandatory but recommended. Standard Edition should only be considered suitable for very small, pilot projects or development environments where scalability, performance, and database size-on-disk are not important considerations. Oracle Database Enterprise Edition, including the Advanced Compression and Partitioning options, is strongly recommended in all other situations.

Refer to My Oracle Support for additional details.

Support for Software Patches and Upgrades

Due to the ongoing nature of software improvement, vendors will issue patches and service packs for the operating systems, application servers and database servers on top of specific versions that Oracle Utilities Customer To Meter has been tested with.

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 Oracle Utilities Customer To Meter production environment.

The exceptions from this rule are Hibernate software version 4.0 GA and Oracle Client version 12.1.0.2. These should not be upgraded.

Always contact Oracle Utilities Customer To Meter 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 of these components is 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
Installing the Version 2.6.0.1.0 Database

This section provides the instructions for installing or upgrading the Oracle Utilities Customer To Meter database. It includes:

- Installation Overview
- Oracle Database Installation
Installation Overview

Refer to Supported Database Platforms for information about the supported platforms on which Oracle Utilities Customer To Meter is verified to operate.

The following types of installation are available for Oracle Utilities Customer To Meter:

- **Initial Install** — a database with no demo data.
- **Upgrade Install** — a database upgrade to version 2.6.0.1.0 from version 2.6.0.0.
- **Demo Install** — a database populated with demo data.

The database installation requires a supported version of the Java Development Kit Version 8.0 and Oracle 12.1.0.2(+) 32-bit client installed on the Windows 64-bit or 32-bit 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 Customer To Meter.

1. Create the database using the Database Configuration Assistant (DBCA). Refer to the article [Master Note: Overview of Database Configuration Assistant (DBCA) (Doc ID 1488770.1)] on My Oracle Support for more information. 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.

2. Enable the mandatory software options.
   - Oracle Spatial OR Oracle Locator
   - Oracle Text

3. Run the following SQL to make sure it is successful.

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

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

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

5. Create the required roles as follows:

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

6. Create the users as follows:

   ```sql
   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 FW43050\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 Customer To Meter 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 example: 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:

```xml
<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
Oracle Database Installation

This section describes how to install the Oracle Database for Oracle Utilities Customer To Meter 2.6.0.1.0. It includes the following:

- Database Scripts and Utilities
- Initial Install or Installing Version 2.6.0.1.0 for the First Time
- Post-Installation Tasks
- Upgrade Install
- Demo Install

**Note:** The installation tools outlined in this guide run on Windows and UNIX/Linux only. Please refer to Supported Database Platforms for more information on supported platforms.

Database Scripts and Utilities

Follow these steps before you begin installing the database:

1. Copy FW-V4.3.0.5.0-Oracle-Database-Multiplatform, CCB-V2.6.0.1.0-FW-Database-PREREQ-MultiPlatform, CCB-V2.6.0.1.0-Oracle-Database-Multiplatform, MDM-V2.2.0-Database and C2M-V2.6.0.1.0-Oracle-Database-Multiplatform directories to your local machine.
   
   These database folders contain several files that will be referred to in the installation process.

2. Set up a Microsoft Windows desktop with the Oracle Client installed.

Initial Install or Installing Version 2.6.0.1.0 for the First Time

This section describes an initial installation of the v2.6.0.1.0 database:

**Note:** You must have a supported version of the Java Development Kit installed on the Windows desktop where you stage and run the database installation package. Refer to the Oracle Utilities Customer To Meter Installation Guide for more information.

This section includes:

- Copying and Decompressing Install Media
- Database Creation
- Installing the CISADM Schema
- Post-Installation Tasks
Copying and Decompressing Install Media
To copy and decompress the Oracle Utilities Customer To Meter database:

1. Download Oracle Utilities Application Framework V4.3.0.5.0 Oracle Database, Oracle Utilities Application Framework V4.3.0.5.0 Database Single Fix Prerequisite Rollup for CCB V2.6.0.1.0, Oracle Utilities Customer Care and Billing v2.6.0.1.0 Oracle Database, Oracle Utilities Meter Data Management V2.2.0.2.0 Oracle Database and Oracle Utilities Customer to Meter V2.6.0.1.0 Oracle Database from the Oracle Software Delivery Cloud.

2. Copy FW-V4.3.0.5.0-Oracle-Database-Multiplatform, CCB-V2.6.0.1.0-FW-Database-PREREQ-MultiPlatform, CCB-V2.6.0.1.0-Oracle-Database-Multiplatform, MDM-V2.2.0-Database and C2M-V2.6.0.1.0-Oracle-Database-Multiplatform directories to your local machine. These files contain all the database components required to install the Oracle Utilities Application Framework and Customer To Meter database.

Database Creation

Note: You must have Oracle Database Server installed on your machine in order to create the database. This step is not required if you are performing a database upgrade from a previous version of Oracle Utilities Customer To Meter.

Creating the Database on UNIX
Create the database using the Database Configuration Assistant (DBCA).

Refer to the article Master Note: Overview of Database Configuration Assistant (DBCA) (Doc ID 1488770.1) on My Oracle Support for more information. Make sure to set character set for database as AL32UTF8.

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 the article Master Note: Overview of Database Configuration Assistant (DBCA) (Doc ID 1488770.1) on My Oracle Support for more information. Make sure to set character set for database as AL32UTF8.

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

Database Globalization Support Consideration
Oracle Utilities Application Framework is a multilingual capable application that supports the storage, processing, and retrieval of data in multiple languages by leveraging the Oracle Database globalization support architecture. Use of the AL32UTF8 Unicode character encoding system allows the database to support multiple languages. If your application supports multiple languages with any one of which being multibyte, then consider the use of Character Length Semantics to store data in database columns in terms of CHARACTERS rather than in terms of BYTEs.

At this time, Oracle Utilities Application Framework only supports CHAR NLS_LENGTH_SEMANTICS setting at the instance level. Since this is an instance wide setting, great care should be taken and a thorough evaluation should be performed
if custom or third party components utilize the same database instance as the Framework application.

**Limitations**
The application will only allow for half of the number of characters if the characters are four bytes. In Java, four-byte characters consume two characters in memory. Due to Legacy Program support and performance considerations, the Framework will allocate storage based on the size defined by the Field.

For example, a Field defined as 12 characters will only be able to store 6 four-byte characters. This does not apply to two- or three-byte characters: in those cases, all 12 of the two- or three-byte characters would fit into the allocated memory. This is not a database limitation - but an application limitation.

MAX_STRING_SIZE of EXTENDED is not supported at this time.

By default, the database is created with BYTE length semantics. To store data using CHARACTER length semantics, follow the procedure below:

**Initial Install**
1. Execute the following statement to set nls_length_semantics=CHAR.
   ```sql
   SQL> ALTER SYSTEM SET nls_length_semantics=CHAR SCOPE=BOTH;
   ```
2. Restart the database.
3. Verify that the nls_length_semantics is CHAR using the following command:
   ```sql
   SQL> SHOW PARAMETER nls_length_semantics
   ```
   **Note:** For pluggable databases ensure to set nls_length_semantics=CHAR.

There are multiple ways to migrate a database from BYTE to CHAR length semantics:

- **By Script:** Refer to the Doc ID 313175.1 on My Oracle Support.
- **Alternative procedure:** The following is an alternate way to create a schema with character-length semantics, and then importing the date from a byte-based export.

**Migrating from BYTE Based Storage to CHARACTER Based Storage**
1. Create a database using DBCA.
2. Execute following statement to set nls_length_semantics=CHAR.
   ```sql
   SQL> ALTER SYSTEM SET nls_length_semantics=CHAR SCOPE=BOTH;
   ```
3. Restart the database.
4. Ensure nls_length_semantics is CHAR using the following command:
   ```sql
   SQL> SHOW PARAMETER nls_length_semantics
   ```
   **Note:** For pluggable database ensure to set nls_length_semantics=CHAR.
5. Export schema from the database that has nls_semantics_length=BYTE.
   ```
   expdp userid=system/<code>@<SID> directory=<DIR_NAME>
   schemas=<schema_name> dumpfile=<schema_name>.dmp
   logfile=<schema_name>.log
   ```

6. Generate DDL from dump file using Oracle impdp utility.
   ```
   impdp userid=system/<code>@<SID> directory=<DIR_NAME>
   DUMPFILE=<schema_name>.dmp SCHEMAS=<schema_name>
   SQLFILE=<schema_name>_DDL.sql
   ```

7. Replace “Byte” with “Char” in <schema_name>_DDL.sql.
   For vi editor (in Linux), use the following command to replace Byte to Char.
   ```
   :%s/BYTE/CHAR/g
   ```

8. Replace the schema name also if it is required for environment.

9. Execute <schema_name>_DDL.sql (generated in step 6) that creates objects in the schema.

   Execute the following command to ensure the number of objects at source and target are equal.
   ```
   SQL>select OWNER || ' ' || OBJECT_TYPE || ' ' || COUNT(*) || ' ' || STATUS FROM DBA_OBJECTS WHERE OWNER in ('<SCHEMA_NAME>') GROUP BY OWNER, OBJECT_TYPE, STATUS ORDER BY OBJECT_TYPE;
   ```

10. If an object is missing for any reason, create it by fixing DDL manually (DDL for each object is available in the file which was created in step 6).

   Execute DDL for the objects that are not created.

11. Generate DDL to disable triggers using following command:
    ```
    SQL> SELECT 'ALTER TABLE' || ' ' || TABLE_NAME || ' ' || 'DISABLE ALL TRIGGERS;' FROM USER_TABLES;
    ```

12. Execute the script generated from step 11 to disable all triggers.

13. Import the data only.

   Use the following command to import data only into the schema created to support CHAR based database storage.
   ```
   impdp userid=system/<code>@<SID> dumpfile=<schema_name>.dmp
   CONTENT=DATA_ONLY SCHEMAS=<schema_name>
   LOGFILE=<schema_name>_import.log
   ```

14. Enable the triggers.

   Use the following command to generate DDL for triggers.
   ```
   SQL>SELECT 'ALTER TABLE' || ' ' || TABLE_NAME || ' ' || 'ENABLE ALL TRIGGERS;' FROM USER_TABLES;
   ```

15. Execute the script generated from step No.14 to enable all triggers.
Installing the CISADM Schema
You will install the Oracle Utilities Application Framework V4.3.0.5.0 Oracle Database, Oracle Utilities Application Framework V4.3.0.5.0 Database Single Fix Prerequisite Rollup for CCB V2.6.0.1.0, Oracle Utilities Customer Care and Billing V2.6.0.1.0 Oracle Database, Oracle Utilities Service and Measurement Data Foundation v2.2.0.2.0, and Oracle Utilities Meter Data Management v2.2.0.2.0 in the same order as mentioned in this document prior to installing Oracle Utilities Customer To Meter 2.6.0.1.0. The files for Oracle Utilities Application Framework installers are located in ..\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\Install-Upgrade folder.

The installation process prompts you for the following information:

- The target database name in which the product is to be installed.
- A database user that will own the application schema (For example: CISADM).
- 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.
- A database user with read-only privileges to the objects in the application schema. (For example: CISREAD).
- A database role that has read-write (select/update/insert/delete) privileges to the objects in the application schema. The application will access the database as this user. (For example: CIS_USER).
- A database role with read-only privileges to the objects in the application schema. (For example: CIS_READ).
- Location for jar files. (The Jar files are bundled with the database package.)
- Java Home (For example: C:\Java\jdk1.8)

Installing the Oracle Utilities Application Framework Database Component Using OraDBI.java
The section below includes the instructions to install the database component.

OraDBI.java is a tool to install and upgrade database components. Before installing the database component, ensure the following prerequisites are met.

- JDK 1.8
- Oracle Database
- Schema (such as CISADM) should exist in the database

To install the Oracle Utilities Application Framework v4.3.0.5.0, follow these steps:

1. Install Framework database component using command prompt utility of Windows from ..\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\Install-Upgrade directory.

2. Prepare the configuration parameters listed below:
   - DB_SERVER – Name of the database server
   - SID- Name of the target database
   - PORT – Port No.
   - DBUSER – Name of the owner of the Database Schema
• DBPASS – Password for the user
• RWUSER – Oracle user with read-write privileges such as CISUSER
• RUSER – Oracle user with read-only privileges such as CISREAD
• RW_USER_ROLE - Oracle database role with read-write privileges such as CIS_USER
• USER_ROLE – Oracle database role with read-only privileges such as CIS_READ
• JAVA_HOME – Location of JDK 1.8 such as C:\Program Files\Java\jdk1.8.0
• CLASS_PATH – Location of Jarfiles such as C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\*

3. There are two options to run OraDBI.java. Choose any of the two options - Using Interactive Mode or Command on Command Line.

• To run OraDBI using Interactive Mode:
  a. Open command prompt / command line on Windows environment.
  b. Set Java Home.

    In the following example, JDK 1.8 is installed at directory C:\Program Files\Java\jdk1.8.0_101.
    SET JAVA_HOME=C:\Program Files\Java\jdk1.8.0_101
  c. Set Class Path.

    In the following example, the required jarfiles (including OraDBI.jar) are available in the directory C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\*
    SET CLASS_PATH= C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\*
  d. Execute the following command at command prompt in Windows environment.

    Using variable paremeters:

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

    Using absolute path of Java home and Jar files:

    "C:\Program Files\Java\jdk1.8.0_101"\bin\java -Xmx1500M -cp
    C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\*
    com.oracle.ouaf.oem.install.OraDBI
  e. The utility prompts you to enter values for the following parameters:
    • Enter the database server hostname:<SERVER NAME>
    • Enter the database port number:<PORT>
    • Enter the database name/SID:<DB NAME>
    • Enter your database username:<CISADM>
Initial Install or Installing Version 2.6.0.1.0 for the First Time

• Enter your password for username CISADM:
• Enter the location for Java Home: <C:\Program Files\Java\jdk1.8.0_101>
• Enter the Oracle user with read-write privileges to Database Schema:<CISUSER>
• Enter the Oracle user with read-only privileges to Database Schema:<CISREAD>
• Enter the database role with read-write privileges to Database Schema:<CIS_USER>
• Enter the database role with read-only privileges to Database Schema:<CIS_READ>
• Enter the name of the target Schema where you want to install or upgrade:<CISADM>
• Enter the password for CISADM schema:

• To run OraDBI using Command on Command Line:
  Run the following command with defined parameters on the command prompt using either absolute value or variable parameters.

  Using variable parameters:
  "%JAVA_HOME"\bin\java -Xmx1500M -cp %CLASS_PATH% com.oracle.ouaf.oem.install.OraDBI -d jdbc:oracle:thin:@<DB_SERVER>:<PORT>;<SID>,<DBUSER>,<DBPASS>,<RW_USER>,<R_USER>,<RW_USER_ROLE>,<R_USER_ROLE>,<DBUSER> -l 1,2 -j <JAVA HOME>

  Using absolute path of Java home and Jar files:
  "C:\Program Files\Java\jdk1.8.0_101"\bin\java -Xmx1500M -cp C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\* com.oracle.ouaf.oem.install.OraDBI -d jdbc:oracle:thin:@<DB_SERVER>:<PORT>;<SID>,<DBUSER>,<DBPASS>,<RW_USER>,<R_USER>,<RW_USER_ROLE>,<R_USER_ROLE>,<DBUSER> -l 1,2 -j "C:\Program Files\Java\jdk1.8.0_101"

  Note: As there is an empty space between the two words ‘Program’ and ‘Files’, it is enclosed in the double quotes in the command above.

This process generates log files in the directory Install-Upgrade\logs.

4. Ensure to check the log files for any errors.

  Note: For OraDBI java, 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.
Installing Prerequisite Database Single Fixes

Before installing any edge products like Oracle Utilities Customer Care and Billing, Service and Measurement Data Foundation, Oracle Utilities Meter Data Management and Oracle Utilities Customer To Meter, you must install Oracle Utilities Framework Prerequisite Database Hot Fixes.

Important!

You MUST execute the C2M_UpdSecFlg.sql file before applying the Framework Database Hot Fixes.

Use the following procedure to execute this file:

1. Open a command prompt.
2. Change directory to ..\C2M-V2.6.0.1.0-Oracle-Database-Multiplatform\C2M\Install-Upgrade.
3. Connect to SQLPLUS as the schema owner.
4. Execute the file as follows:
   
   @C2M_UpdSecFlg.sql

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

Applying Hot Fixes from a Windows Machine

Note: Java 8 JDK should be installed on the machine to use the command. Ensure to install the JDK that is supported for your platform.

1. Copy the ..\CCB-V2.6.0.1.0-FW-Database-PREREQ-MultiPlatform\FW43050-HFix\db_patch_standalone.jar to a directory on Windows, under c:\dbpatch_tools and extract the db_patch_standalone.jar using below 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 ..\CCB-V2.6.0.1.0-FW-Database-PREREQ-MultiPlatform\FW43050-HFix 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>

Applying Hotfixes from a Unix Standalone Server

Note: Java 8 JDK should be installed on the machine to use the command. Ensure to install the JDK that is supported for your platform.
1. Copy the ..\CCB-V2.6.0.1.0-FW-Database-PREREQ-MultiPlatform\FW43050-HFix\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 ..\CCB-V2.6.0.1.0-FW-Database-PREREQ-MultiPlatform\FW43050-HFix 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>

**Installing the Oracle Utilities Customer Care and Billing Database Component Using OraDBI.java**

OraDBI.java is a new tool to install and upgrade database components. Before installing the Oracle Utilities Customer Care and Billing v2.6.0.1.0, ensure the following prerequisites are met.

- JDK 1.8
- Oracle Database
- Schema (such as CISADM) should exist in the database

To install the Oracle Utilities Customer Care and Billing v2.6.0.1.0 database, follow these steps:

1. Install the Customer Care and Billing database component using the command prompt utility of Windows from ..\CCB-V2.6.0.1.0-Oracle-Database-Multiplatform\CCB\Install-Upgrade\.

2. Prepare the configuration parameters listed below:
   
   - DB_SERVER – Name of the database server
   - SID- Name of the target database
   - PORT – Port No.
   - DBUSER – Name of the owner of the Database Schema
   - DBPASS – Password for the user
   - RWUSER – Oracle user with read-write privileges such as CISUSER
   - RUSER – Oracle user with read-only privileges such as CISREAD
   - RW_USER_ROLE - Oracle database role with read-write privileges such as CIS_USER
   - USER_ROLE – Oracle database role with read-only privileges such as CIS_READ
- JAVA_HOME – Location of JDK 1.8 such as C:\Program Files\Java\jdk1.8.0
- CLASS_PATH – Location of Jarfiles such as C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\*

3. There are two options to run OraDBI.java. Choose any of the two options - Using Interactive Mode or Command on Command Line.

- **To run OraDBI using Interactive Mode:**
  a. Open command prompt / command line on Windows environment.
  b. Set Java Home.

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

    SET JAVA_HOME=C:\Program Files\Java\jdk1.8.0_101
  c. Set Class Path.

    In the following example, the required jarfiles (including OraDBI.jar) are available in the directory C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\*.

    SET CLASS_PATH= C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\*
  d. Execute the following command at command prompt in Windows environment.

    Using variable parameters:

    
    "%JAVA_HOME%"\bin\java -Xmx1500M -cp %CLASS_PATH% com.oracle.ouaf.oem.install.OraDBI
    
    Using absolute path of Java home and Jar files:

    "C:\Program Files\Java\jdk1.8.0_101"\bin\java -Xmx1500M -cp C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\* com.oracle.ouaf.oem.install.OraDBI

e. The utility prompts you to enter values for the following parameters:

- Enter the database server hostname:<SERVER NAME>
- Enter the database port number:<PORT>
- Enter the database name/SID:<DB NAME>
- Enter your database username:<CISADM>
- Enter your password for username CISADM:
- Enter the location for Java Home: <C:\Program Files\Java\jdk1.8.0_101>
- Enter the Oracle user with read-write privileges to Database Schema:<CISUSER>
- Enter the Oracle user with read-only privileges to Database Schema:<CISREAD>
• Enter the database role with read-write privileges to Database Schema:<CIS_USER>
• Enter the database role with read-only privileges to Database Schema:<CIS_READ>
• Enter the name of the target Schema where you want to install or upgrade:<CISADM>
• Enter the password for CISADM schema:

**To run OraDBI using Command on Command Line:**

Run the following command with defined parameters on the command prompt using either absolute value or variable parameters.

**Using variable parameters:**

```
"%JAVA_HOME\"\bin\java -Xmx1500M -cp %CLASS_PATH% com.oracle.ouaf.oem.install.OraDBI -d jdbc:oracle:thin:@<DB_SERVER>:<PORT>/
<SID>,<DBUSER>,<DBPASS>,<RW_USER>,<R_USER>,<RW_USER_ROLE>,<R_USER_ROLE>,<DBUSER> -l 1,2 -j <JAVA_HOME>
```

**Using absolute path of Java home and Jar files:**

```
"C:\Program Files\Java\jdk1.8.0_101\"\bin\java -Xmx1500M -cp C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\*
<SID>,<DBUSER>,<DBPASS>,<RW_USER>,<R_USER>,<RW_USER_ROLE>,<R_USER_ROLE>,<DBUSER> -l 1,2 -j "C:\Program Files\Java\jdk1.8.0_101"
```

**Note:** As there is an empty space between the two words ‘Program’ and ‘Files’, it is enclosed in the double quotes in the command above.

This process generates log files in the directory Install-Upgrade\logs.

4. Ensure to check the log files for any errors.

**Note:** For OraDBI java, 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.
```

**Installing the Oracle Utilities Service and Measurement Data Foundation Component Using OraDBI.java**

OraDBI.java is a tool used to install and upgrade database components. Before installing the Oracle Utilities Service and Measurement v2.2.0.2.0, ensure the following prerequisites are met.

- JDK 1.8
- Oracle Database
To install the Oracle Utilities Service and Measurement v2.2.0.2.0, follow these steps:

1. Copy MDM-V2.2.0-Database.zip directory to your local machine.
   - This file contains all the database components required to install the Oracle Utilities Service and Measurement Data Foundation database.

2. Install the Service and Measurement Data Foundation database component using command prompt utility of Windows from ..\MDM-V2.2.0-Database\D1\Install-Upgrade.

3. Prepare the configuration parameters listed below:
   - DB_SERVER - Name of the database server
   - SID - Name of the target database
   - PORT - Port No.
   - DBUSER - Name of the owner of the Database Schema
   - DBPASS - Password for the user
   - RWUSER - Oracle user with read-write privileges such as CISUSER
   - RUSER - Oracle user with read-only privileges such as CISREAD
   - RW_USER_ROLE - Oracle database role with read-write privileges such as CIS_USER
   - USER_ROLE - Oracle database role with read-only privileges such as CIS_READ
   - JAVA_HOME - Location of JDK 1.8 such as C:\Program Files\Java\jdk1.8.0
   - CLASS_PATH - Location of jarfiles such as C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\*

4. There are two options to run OraDBI.java. Choose any of the two options - Using Interactive Mode or Command on Command Line.
   - **To run OraDBI using Interactive Mode:**
     a. Open command prompt / command line on Windows environment.
     b. Set Java Home.
        - In the following example, JDK 1.8 is installed at directory C:\Program Files\Java\jdk1.8.0_101.
        ```plaintext
        SET JAVA_HOME=C:\Program Files\Java\jdk1.8.0_101
        ```
     c. Set Class Path.
        - In the following example, the required jarfiles (including OraDBI.jar) are available in the directory C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\*.
        ```plaintext
        SET CLASS_PATH= C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\*
        ```
     d. Execute the following command at command prompt in Windows environment.
Using variable parameters:
"%JAVA_HOME%"\bin\java -Xmx1500M -cp %CLASS_PATH%
com.oracle.ouaf.oem.install.OraDBI

Using absolute path of Java home and Jar files:
"C:\Program Files\Java\jdk1.8.0_101"\bin\java -Xmx1500M -cp
C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles* com.oracle.ouaf.oem.install.OraDBI

e. The utility prompts you to enter values for the following parameters:

• Enter the database server hostname:<SERVER NAME>
• Enter the database port number:<PORT>
• Enter the database name/SID:<DB NAME>
• Enter your database username:<CISADM>
• Enter your password for username CISADM:
• Enter the location for Java Home: <C:\Program
Files\Java\jdk1.8.0_101>
• Enter the Oracle user with read-write privileges to Database
Schema:<CISUSER>
• Enter the Oracle user with read-only privileges to Database
Schema:<CISREAD>
• Enter the database role with read-write privileges to Database
Schema:<CIS_USER>
• Enter the database role with read-only privileges to Database
Schema:<CIS_READ>
• Enter the name of the target Schema where you want to install or
upgrade:<CISADM>
• Enter the password for CISADM schema:

• To run OraDBI using Command on Command Line:

Run the following command with defined parameters on the command prompt
using either absolute value or variable parameters.

Using variable parameters:
"%JAVA_HOME%"\bin\java -Xmx1500M -cp %CLASS_PATH%
jdbc:oracle:thin:@<DB_SERVER>:<PORT>:
<DB_NAME>,<DBUSER>,<DBPASS>,<RW_USER>,<R_USER>,<RW_USER_ROLE>,<R_USER_ROLE>,<DBUSER> -l 1,2 -j <JAVA_HOME>

Using absolute path of Java home and Jar files:
"C:\Program Files\Java\jdk1.8.0_101"\bin\java -Xmx1500M -cp
C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles* jdbc:oracle:thin:@<DB_SERVER>:<PORT>:
<DB_NAME>,<DBUSER>,<DBPASS>,<RW_USER>,<R_USER>,<RW_USER_ROLE>,<R_USER_ROLE>,<DBUSER> -l 1,2 -j "C:\Program Files\Java\jdk1.8.0_101"
Note: As there is an empty space between the two words 'Program' and 'Files', it is enclosed in the double quotes in the command above.

This process generates log files in the directory Install-Upgrade\logs.

5. Ensure to check the log files for any errors.

Note: For OraDBI java, 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.
```

**Installing the Oracle Utilities Meter Data Management Component Using OraDBI.java**

OraDBI.java is a tool used to install and upgrade database components. Before installing the Oracle Utilities Meter Data Management v2.2.0.2.0, ensure the following prerequisites are met.

- JDK 1.8
- Oracle Database
- Schema (such as CISADM) should exist in the database

To install the Oracle Utilities Meter Data Management v2.2.0.2.0, follow these steps:

1. Copy MDM-V2.2.0-Database.zip directory to your local machine.
   
   This file contains all the database components required to install the Oracle Utilities Meter Data Management database.

2. Install the Meter Data Management database component using command prompt utility of Windows from ..\MDM-V2.2.0-Database\D2\Install-Upgrade.

3. Prepare the configuration parameters listed below:

   - **DB_SERVER** - Name of the database server
   - **SID** - Name of the target database
   - **PORT** - Port No.
   - **DBUSER** - Name of the owner of the Database Schema
   - **DBPASS** - Password for the user
   - **RWUSER** - Oracle user with read-write privileges such as CISUSER
   - **RUSER** - Oracle user with read-only privileges such as CISREAD
   - **RW_USER_ROLE** - Oracle database role with read-write privileges such as CIS_USER
   - **USER_ROLE** - Oracle database role with read-only privileges such as CIS_READ
   - **JAVA_HOME** - Location of JDK 1.8 such as C:\Program Files\Java\jdk1.8.0
4. There are two options to run OraDBI.java. Choose any of the two options - Using Interactive Mode or Command on Command Line.

**To run OraDBI using Interactive Mode:**

a. Open command prompt / command line on Windows environment.

b. Set Java Home.

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

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

   c. Set Class Path.

   In the following example, the required jarfiles (including OraDBI.jar) are available in the directory C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\*.

   ```
   SET CLASS_PATH=C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\*
   ```

d. Execute the following command at command prompt in Windows environment.

   Using variable parameters:

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

   Using absolute path of Java home and Jar files:

   ```
   "C:\Program Files\Java\jdk1.8.0_101"\bin\java -Xmx1500M -cp C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\* com.oracle.ouaf.oem.install.OraDBI
   ```

e. The utility prompts you to enter values for the following parameters:

   - Enter the database server hostname:<SERVER NAME>
   - Enter the database port number:<PORT>
   - Enter the database name/SID:<DB NAME>
   - Enter your database username:<CISADM>
   - Enter your password for username CISADM:
   - Enter the location for Java Home: <C:\Program Files\Java\jdk1.8.0_101>
   - Enter the Oracle user with read-write privileges to Database Schema:<CISUSER>
   - Enter the Oracle user with read-only privileges to Database Schema:<CISREAD>
   - Enter the database role with read-write privileges to Database Schema:<CIS_USER>
   - Enter the database role with read-only privileges to Database Schema:<CIS_READ>
• Enter the name of the target Schema where you want to install or upgrade:<CISADM>
• Enter the password for CISADM schema:

**To run OraDBI using Command on Command Line:**

Run the following command with defined parameters on the command prompt using either absolute value or variable parameters.

**Using variable parameters:**

```bash
"%JAVA_HOME%"\bin\java -Xmx1500M -cp %CLASS_PATH%
com.oracle.ouaf.oem.install.OraDBI -d
jdbc:oracle:thin:@<DB_SERVER>:<PORT>:/
<SID>,<DBUSER>,<DBPASS>,<RW_USER>,<R_USER>,<RW_USER_ROLE>,<R_USER_ROLE>,<DBUSER> -l 1,2 -j <JAVA_HOME>
```

**Using absolute path of Java home and Jar files:**

```bash
"C:\Program Files\Java\jdk1.8.0_101"\bin\java -Xmx1500M -cp C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW.jarfiles:\
com.oracle.ouaf.oem.install.OraDBI -d
jdbc:oracle:thin:@<DB_SERVER>:<PORT>:/
<SID>,<DBUSER>,<DBPASS>,<RW_USER>,<R_USER>,<RW_USER_ROLE>,<R_USER_ROLE>,<DBUSER> -l 1,2 -j "C:\Program Files\Java\jdk1.8.0_101"
```

**Note:** As there is an empty space between the two words 'Program' and 'Files', it is enclosed in the double quotes in the command above.

This process generates log files in the directory Install-Upgrade\logs.

5. Ensure to check the log files for any errors.

Note: For OraDBI java, 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.

**Optional:** This optional step should be executed if you have installed Oracle Utilities Meter Data Analytics 2.5.0.0.2 (2.5 Patch Set 2), or if you plan to install it in the future.

Navigate to ..\C2M\V2.6.0.1.0\MDM-V2.2.0-Database\D2\Post-Upgrade folder and run Materialized_View_Creation.sql from sql prompt as follows:

a. Connect to Database Owner Schema. (for example: <CISADM>/
   <CISADM>@<SERVICE_NAME>)

b. Run Materialized_View_Creation.sql as @Materialized_View_Creation.sql from
   sql prompt.

After the required changes are complete, configure security by following the steps in the Configuring Security section.
Installing the Oracle Utilities Customer To Meter Component Using OraDBI.java

OraDBI.java is a tool used to install and upgrade database components. Before installing the Oracle Utilities Customer To Meter v2.6.0.1.0, ensure the following prerequisites are met.

- JDK 1.8
- Oracle Database
- Schema (such as CISADM) should exist in the database

To install the Oracle Utilities Customer To Meter v2.6.0.1.0, follow these steps:

1. Copy C2M-V2.6.0.1.0-Database.zip directory to your local machine.
   This file contains all the database components required to install the Oracle Utilities Customer To Meter database.

2. Install the Customer To Meter database component using command prompt utility of Windows from ..\C2M-V2.6.0.1.0-Oracle-Database-Multiplatform\C2M\Install-Upgrade.

3. Prepare the configuration parameters listed below:
   - DB_SERVER - Name of the database server
   - SID - Name of the target database
   - PORT - Port No.
   - DBUSER - Name of the owner of the Database Schema
   - DBPASS - Password for the user
   - RWUSER - Oracle user with read-write privileges such as CISUSER
   - RUSER - Oracle user with read-only privileges such as CISREAD
   - RW_USER_ROLE - Oracle database role with read-write privileges such as CIS_USER
   - USER_ROLE - Oracle database role with read-only privileges such as CIS_READ
   - JAVA_HOME - Location of JDK 1.8 such as C:\Program Files\Java\jdk1.8.0
   - CLASS_PATH - Location of Jarfiles such as C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\*

4. There are two options to run OraDBI.java. Choose any of the two options - Using Interactive Mode or Command on Command Line.

   - **To run OraDBI using Interactive Mode:**
     a. Open command prompt / command line on Windows environment.
     b. Set Java Home.
        In the following example, JDK 1.8 is installed at directory C:\Program Files\Java\jdk1.8.0_101.
        SET JAVA_HOME=C:\Program Files\Java\jdk1.8.0_101
     c. Set Class Path.
In the following example, the required jar files (including OraDBI.jar) are available in the directory C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles*.

SET CLASS_PATH = C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles*.

d. Execute the following command at command prompt in Windows environment.

Using variable parameters:
"%JAVA_HOME%\bin\java -Xmx1500M -cp %CLASS_PATH% com.oracle.ouaf.oem.install.OraDBI

Using absolute path of Java home and Jar files:
"C:\Program Files\Java\jdk1.8.0_101\bin\java -Xmx1500M -cp C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles* com.oracle.ouaf.oem.install.OraDBI

e. The utility prompts you to enter values for the following parameters:

• Enter the database server hostname:<SERVER NAME>
• Enter the database port number:<PORT>
• Enter the database name/SID:<DB NAME>
• Enter your database username:<CISADM>
• Enter your password for username CISADM:
• Enter the location for Java Home: <C:\Program Files\Java\jdk1.8.0_101>
• Enter the Oracle user with read-write privileges to Database Schema:<CISUSER>
• Enter the Oracle user with read-only privileges to Database Schema:<CISREAD>
• Enter the database role with read-write privileges to Database Schema:<CIS_USER>
• Enter the database role with read-only privileges to Database Schema:<CIS_READ>
• Enter the name of the target Schema where you want to install or upgrade:<CISADM>
• Enter the password for CISADM schema:

• To run OraDBI using Command on Command Line:

Run the following command with defined parameters on the command prompt using either absolute value or variable parameters.

Using variable parameters:
"%JAVA_HOME%\bin\java -Xmx1500M -cp %CLASS_PATH% com.oracle.ouaf.oem.install.OraDBI -d jdbc:oracle:thin:@<DB_SERVER>:<PORT>/<SID>,<DBUSER>,<DBPASS>,<RW_USER>,<R_USER>,<RW_USER_ROLE>,<R_USER_ROLE>,<DBUSER> -l 1,2 -j <JAVA HOME>
Using absolute path of Java home and Jar files:

"C:\Program Files\Java\jdk1.8.0_101\bin\java -Xmx1500M -cp C:\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\jarfiles\ com.oracle.ouaf.oem.install.OraDBI -d jdbc:oracle:thin:@<DB_SERVER>:<PORT>/<SID>,<DBUSER>,<DBPASS>,<RW_USER>,<R_USER>,<RW_USER_ROLE>,<R_USER_ROLE>,<DBUSER> -l 1,2 -j "C:\Program Files\Java\jdk1.8.0_101"

Note: As there is an empty space between the two words 'Program' and 'Files', it is enclosed in the double quotes in the command above.

This process generates log files in the directory Install-Upgrade\logs.

5. Ensure to check the log files for any errors.

Note: For OraDBI java, 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.

```
...
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.
```

Installing the Oracle Utilities Smart Grid Gateway Component Using OraDBI.java (Optional)

Installation of Oracle Utilities Smart Grid Gateway is optional, and requires a separate license.

Note: When implementing Oracle Utilities Smart Grid Gateway with Oracle Utilities Customer To Meter, both the Smart Grid Gateway and Meter Data Management database components should be installed in the same database.

To install the Oracle Utilities Smart Grid Gateway Database Component:

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

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

• Using the Interactive Mode
• Using the Command Line Mode

Using the Interactive Mode
The following procedure lists the steps to install the Oracle Utilities Smart Grid Gateway component using OraDBI.

Run the following command with the defined parameters on the command prompt from the ..\DB\SGG.V2.2.0.2.0\Install-Upgrade\ directory.

1. Open a command line prompt.
2. Set Java Home.
   In the following example, JDK 1.8 is installed in the C:\Program Files\Java\jdk1.8.0 directory.
   
   SET JAVA_HOME=C:\Program Files\Java\jdk1.8.0

3. Set the class path.
   
   SET CLASS_PATH=C:\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"\bin\java -Xmx1500M -cp C:\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: <SERVICE_NAME>
   • 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.


Using the Command Line Mode

Run the following command with the defined parameters on the command prompt from ..\DB\SGG.V2.2.0.2.0\Install-Upgrade\ directory.

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


If you chose to continue, OraDBI first checks for the existence of each of the users specified and prompts for their password, default tablespace, and temporary tablespace.
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.

- OraDBI can be executed by a non-schema owner.

Post-Installation Tasks

- Enable USER_LOCK Package

- Generating Database Statistics

- Creating Activity Statistics Materialized View

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:
   ```sql
   @?/rdbms/admin/userlock.sql
   ```
3. Grant permission by running the following SQL:
   ```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.
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.

Creating Activity Statistics Materialized View
To improve the performance of drill down queries, use the following procedure to create the materialized view and then refresh it.

Navigate to ..\C2M\V2.6.0.1.0\MDM-V2.2.0-Database\D1\Post-Upgrade and run the scripts below.
1. Login as CISADM user.
2. At the SQL prompt, run the following:
   @D1_ACTIVITY_STAT_MV.sql
   @D1_MV_REFRESH_PROC.sql

Upgrade Install
This section describes how to upgrade the database components for Oracle Utilities Customer To Meter, including:
- Copying and Decompressing Install Media
- Upgrading the CISADM Schema to Version 2.6.0.1.0

Copying and Decompressing Install Media
To copy and decompress the Oracle Utilities Customer To Meter database:
1. Download Oracle Utilities Application Framework V4.3.0.5.0 Oracle Database,
   Oracle Utilities Application Framework V4.3.0.5.0 Database Single Fix Prerequisite Rollup for CCB V2.6.0.1.0, Oracle Utilities Customer Care and Billing v2.6.0.1.0
   Oracle Database, Oracle Utilities Meter Data Management V2.2.0.2.0 Oracle Database and Oracle Utilities Customer to Meter V2.6.0.1.0 Oracle Database from
   the Oracle Software Delivery Cloud.
2. Copy FW-V4.3.0.5.0-Oracle-Database-Multiplatform, CCB-V2.6.0.1.0-FW-
   Database-PREREQ-MultiPlatform, CCB-V2.6.0.1.0-Oracle-Database-
   Multiplatform, MDM-V2.2.0-Database and C2M-V2.6.0.1.0-Oracle-Database-
   Multiplatform directories to your local machine.
   These files contain all the database components required to install the Oracle Utilities Application Framework and Customer To Meter database.

Upgrading the CISADM Schema to Version 2.6.0.1.0
This section assumes an existing Oracle Utilities Customer To Meter on top of Oracle Utilities Application Framework. The following upgrade paths are described:
- Upgrading from Version 2.6.0 to 2.6.0.1.0
Upgrading from Version 2.6.0 to 2.6.0.1.0
You must install the Oracle Utilities Application Framework version 4.3.0.5.0, Oracle Utilities Application Framework V4.3.0.5.0 Database Single Fix Prerequisite Rollup for CCB V2.6.0.1.0, Oracle Utilities Customer Care and Billing 2.6.0.1.0, Oracle Utilities Service and Measurement v2.2.0.2.0, Oracle Utilities Meter Data Management v2.2.0.2.0, and Oracle Utilities Customer To Meter V2.6.0.1.0. The files for Oracle Utilities Application Framework installer is located in ..\FW-V4.3.0.5.0-Oracle-Database-Multiplatform\FW\Install-Upgrade folder.

Upgrading the Database as Non-Schema Owner
The product allows Non-Schema owners to run the database upgrade.

To perform upgrade, the non-schema owner must have the following database grants:

- grant connect, CREATE SESSION to <Non-Schema owner>;
- grant select on <Schema owner>.CI_WFM to <Non-Schema owner>;
- grant select on <Schema owner>.CI_WFM_OPT to <Non-Schema owner>;

Installing the Oracle Utilities Application Framework Database Component
For instructions, refer to Installing the Oracle Utilities Application Framework Database Component Using OraDBI.java on page 2-8.

Installing Prerequisite Database Single Fixes
For instructions, refer to Installing Prerequisite Database Single Fixes on page 2-11.

Installing the Oracle Utilities Customer Care and Billing Database Component
For instructions, refer to Installing the Oracle Utilities Customer Care and Billing Database Component Using OraDBI.java on page 2-12.

Installing the Oracle Utilities Service and Measurement Data Foundation Database Component
For instructions, refer to Installing the Oracle Utilities Service and Measurement Data Foundation Component Using OraDBI.java on page 2-14.

Installing the Oracle Utilities Meter Data Management Database Component
For instructions, refer to Installing the Oracle Utilities Meter Data Management Component Using OraDBI.java on page 2-17.

Installing the Oracle Utilities Customer To Meter Database Component
For instructions, refer to Installing the Oracle Utilities Customer To Meter Component Using OraDBI.java on page 2-20.
Demo Install

This section describes how to install the demo database components for Oracle Utilities Customer To Meter, including:

- Copying and Decompressing Install Media
- Creating the Database
- Importing the Demo Dump File
- Configuring Security

Copying and Decompressing Install Media

To copy and decompress the Oracle Utilities Customer To Meter database:

1. Download the Oracle Utilities Customer To Meter V2.6.0.1.0 Oracle database from the Oracle Software Delivery Cloud.
2. Copy the C2M-V2.6.0.1.0-Oracle-Database-MultiplatForm directory to your local machine. The database folder contains several files that will be referred to in the installation process.

Creating the Database

Note: You must have Oracle Database Server 12.1.0.2+ installed on your machine in order to create 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 the article Master Note: Overview of Database Configuration Assistant (DBCA) (Doc ID 1488770.1) on My Oracle Support for more information. Make sure to set character set for database as AL32UTF8.

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 the article Master Note: Overview of Database Configuration Assistant (DBCA) (Doc ID 1488770.1) on My Oracle Support for more information. Make sure to set character set for database as AL32UTF8.

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

Database Storage BYTES / CHARACTER

Database created by default will store data in BYTES. To store data in CHARACTER follow the procedure below:

Initial Install

1. Execute the following statement to set nls_length_semantics=CHAR.
   
   SQL> ALTER SYSTEM SET nls_length_semantics=CHAR SCOPE=BOTH;
2. Restart the database.
3. Ensure nls_length_semantics is CHAR. Perform the following command:
   ```sql
   SQL> SHOW PARAMETER nls_length_semantics
   ```

   **Note:** For pluggable database ensure to set
   nls_length_semantics=CHAR for both container and pluggable database.

---

**Upgrade and Migration from BYTE Based Storage to CHARACTER Based Storage**

1. Create a database using DBCA.
2. Execute following statement to set nls_length_semantics=CHAR.
   ```sql
   SQL> ALTER SYSTEM SET nls_length_semantics=CHAR SCOPE=BOTH;
   ```
3. Restart the database.
4. Ensure nls_length_semantics is CHAR using the following command:
   ```sql
   SQL> SHOW PARAMETER nls_length_semantics
   ```

   **Note:** For pluggable database ensure to set
   nls_length_semantics=CHAR for container and pluggable database both.

5. Export schema from the database that has nls_semantics_length=BYTE.
   ```bash
   expdp userid=system/<code>@<SID> directory=<DIR_NAME> schemas=<schema_name> dumpfile=<schema_name>.dmp logfile=<schema_name>.log
   ```

6. Generate DDL from dump file using Oracle impdp utility.
   ```bash
   impdp userid=system/<code>@<SID> directory=<DIR_NAME> DUMPFILE=<schema_name>.dmp SCHEMAS=<schema_name> SQLFILE=<schema_name>_DDL.sql
   ```

7. Replace “Byte” with “Char” in <schema_name>DDL.sql.
   For vi editor (in Linux), use the following command to replace Byte to Char.
   ```bash
   :%s/BYTE/CHAR/g
   ```

8. Replace the schema name also if it is required for environment.
9. Execute <schema_name>DDL.sql (generated in step 6) that creates objects in the schema.

   Execute the following command to ensure the number of objects at source and target are equal.
   ```sql
   SQL>select OWNER || '  ' || OBJECT_TYPE || '  ' || COUNT(*) || '  ' || STATUS FROM DBA_OBJECTS WHERE OWNER in ('<SCHEMA_NAME>') GROUP BY OWNER, OBJECT_TYPE , STATUS ORDER BY OBJECT_TYPE;
   ```

10. If an object is missing for any reason, create it by fixing DDL manually (DDL for each object is available in the file which was created in step 6).

    Execute DDL for the objects that are not created.
11. Generate DDL to disable triggers using following command:

   SQL> SELECT 'ALTER TABLE' || ' ' || TABLE_NAME || ' ' || 'DISABLE ALL TRIGGERS;' FROM USER_TABLES;

12. Execute the script generated from step 11 to disable all triggers.

13. Import the data only.

   Use the following command to import data only into the schema created to support CHAR based database storage.

   impdp userid=system/<code>@<SID> dumpfile=<schema_name>.dmp CONTENT=DATA_ONLY SCHEMAS=<schema_name> LOGFILE=<schema_name>_import.log

14. Enable the triggers.

   Use the following command to generate DDL for triggers.

   SQL> SELECT 'ALTER TABLE' || ' ' || TABLE_NAME || ' ' || 'ENABLE ALL TRIGGERS;' FROM USER_TABLES;

15. Execute the script generated from step No.14 to enable all triggers.

**Importing the 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 demo dump:

   NOTE: Ensure the ..\C2M-V2.6.0.1.0-Oracle-Database-Multiplatform\C2M\Demo/exp_demo.dmp.gz file is extracted and available in data_pump_dir's location before running the below import command.

   impdp directory= data_pump_dir dumpfile= exp_demo.dmp logfile=exp_demo.log schemas=CISADM

**Configuring Security**

The configuration utility and scripts are located in the ..\C2M-V2.6.0.1.0-Oracle-Database-Multiplatform\C2M\Security folder. You can run this utility from a Linux or a Windows machine.

To configure security for all objects from a Linux Standalone machine, you can either set the parameters as describe in Linux Option 1 or using absolute value as describe in Linux Option 2.

**Linux OPTION 1: Using Variables**

1. Set the following parameters:

   a. JAVA_HOME - location of JDK 1.8

      export JAVA_HOME=/scratch/software/jdk1.8.0_102

   b. PATH - location of JDK bin

      export PATH=$JAVA_HOME/bin:$PATH
c. CLASS_PATH - location of jarfiles. This can be found in ..\..\CCB-V2.6.0.1.0-Oracle-Database-MultiplatForm\CCB\Security\lib
   export CLASS_PATH=..\..\CCB-V2.6.0.1.0-Oracle-Database-MultiplatForm\CCB\Security\lib\*

d. DB_SERVER - database server
   export DB_SERVER=<database server>

e. PORT - database port number
   export PORT=<port number>

f. SID - database name
   export SID=<database name>

g. DBUSER - database username such as CISADM
   export DBUSER=<CISADM>

h. DBPASS - database password for username
   export DBPASS=<password>

i. RW_USER - database username with read-write privileges such as CISUSER
   export RW_USER=<CISUSER>

j. R_USER - database username with read privileges such as CISREAD
   export R_USER=<CISREAD>

k. RW_USER_ROLE - database role with read-write privileges such as CIS_USER
   export RW_USER_ROLE=<CIS_USER>

l. R_USER_ROLE - database role with read privileges such as CIS_READ
   export R_USER_ROLE=<CIS_READ>

2. Execute the OraGenSec utility using below command.
   $JAVA_HOME/bin/java $CLASS_PATH
   com.oracle.ouaf.oem.install.OraGenSec -d
   $DBUSER,$DBPASS,jdbc:oracle:thin:@$DB_SERVER:$PORT/$SID -a A -r
   $RW_USER_ROLE,$R_USER_ROLE -u $RWUSER,$RUSER

Linux OPTION 2: Using Absolute Value

1. Set the CLASSPATH parameter
   export CLASSPATH = ../..\CCB-V2.6.0.1.0-Oracle-Database-MultiplatForm/CCB/Security/lib

2. Execute OraGenSec utility using below command just replace with the actual values.
   /scratch/software/jdk1.8.0_102/bin/java com.oracle.ouaf.oem.install.OraGenSec -d
   $DBUSER,<DBPASS>,jdbc:oracle:thin:@<DB_SERVER>:<PORT>/:<SID> -a A -r
   $RW_USER_ROLE,$R_USER_ROLE -u $RWUSER,$RUSER

To configure security for all objects from a Windows machine, you can either set the parameters as describe in Windows Option 1 or using absolute value as describe in Windows Option 2.
Windows OPTION 1: Using Variables

1. Set the following parameters:
   a. JAVA_HOME - location of JDK 1.8
      
      set JAVA_HOME=C:\Software\Java\jre1.8.0_91
   b. PATH - location of JDK bin
      
      set PATH=%JAVA_HOME%/bin;%PATH
   c. CLASS_PATH - location of jarfiles. This can be found in ..\..\CCB-V2.6.0.1.0-Oracle-Database-MultiplatForm\CCB\Security\lib
      
      set CLASS_PATH=..\..\CCB-V2.6.0.1.0-Oracle-Database-MultiplatForm\CCB\Security\lib\*
   d. DB_SERVER - database server
      
      set DB_SERVER=<database server>
   e. PORT - database port number
      
      set PORT=<port number>
   f. SID - database name
      
      set SID=<database name>
   g. DBUSER - database username such as CISADM
      
      set DBUSER=<CISADM>
   h. DBPASS - database password for username
      
      set DBPASS=<password>
   i. RW_USER - database username with read-write privileges such as CISUSER
      
      set RW_USER=<CISUSER>
   j. R_USER - database username with read privileges such as CISREAD
      
      set R_USER=<CISREAD>
   k. RW_USER_ROLE - database role with read-write privileges such as CIS_USER
      
      set RW_USER_ROLE=<CIS_USER>
   l. R_USER_ROLE - database role with read privileges such as CIS_READ
      
      set R_USER_ROLE=<CIS_READ>

2. Execute the OraGenSec utility using below command.

   "%JAVA_HOME%"\bin\java -Xmx1500M -cp %CLASS_PATH% com.oracle.ouaf.oem.install.OraGenSec -d %DBUSER%,%DBPASS%,jdbc:oracle:thin:@%DB_SERVER%:%PORT%/%SID% -a A -r %RW_USER_ROLE%,%R_USER_ROLE% -u %RWUSER%,%RUSER% 

Windows OPTION 2: Using Absolute Value

1. Execute OraGenSec utility using below command just replace with the actual values.

   <JAVA_HOME>\bin\java -Xmx1500M -cp <CLASS_PATH>* com.oracle.ouaf.oem.install.OraGenSec -d
<DBUSER>,<DBPASS>;jdbc:oracle:thin:@<DB_SERVER>:<PORT>/<SID> -a A -r <RW_USER>,<R_USER> -u <RW_USER_ROLE>,<R_USER_ROLE>

Note: Database vault must be disabled before running.

The utility configures security for the application owner schema objects.

Oragensec by default grant permissions to CIS_USER and CIS_READ Role. If you prefer to use site-specific roles then execute Oragensec after providing command line options and specifying the specific roles.

OraGenSec Java Usage:

```
[-u <arg>]
```

OraGenSec Help:

- `-a <arg>`: generate security for All objects in the database
- `-d <arg>`: db connection as: db_host:db_port/db_service
- `-f <arg>`: generate security for specific objects from an input File
- `-h`: help
- `-l <arg>`: log file
- `-o <arg>`: generate security for comma separated list of objects
- `-q`: silent mode
- `-r <arg>`: roles as: CIS_READ,CIS_USER
- `-u <arg>`: comma-separated list of users to create synonyms for
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 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_.

Oracle Utilities Customer To Meter Database Administrator’s Guide
**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
• XT206C2
• 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:

```xml
<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:

```xml
<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:
   
   \[ \text{OF}]\text{C/M/T}\text{NNN\_SEQ} 
   
   • OF stands for Owner Flag. For example, for Framework its F1 and for CCB it is
     C1 and for MDM it is D1.
   
   • 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:
   
   \[ \text{OF}]\text{C/M/T}\text{NNN\_Column\_Name\_SEQ} 
   
   • 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.
   
   \[ \text{OF}\text{Column\_Name\_SEQ} 
   
   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.
XML Type Support

The product supports XML Type. XML Type provides the 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 user-defined keys, the KEY_VALIDATION_FLG should be set as 'ALL'. This checks the existence of the key before inserting a new one.

Table Classification and Table Volume Flags

There are multiple types of tables in the application, namely Admin system tables, Admin non-system tables, master tables and transaction tables. The Table Classification flag (TBL_CLASSIFICATION_FLG) sets the appropriate value for this lookup field to give a better view of the table classification.

Table Volume flag (TBL_VOLUME_FLG) is a customer modifiable field which is initially populated by the product, but can be overridden by implementation. The field gives an idea of the relative data volume (categorized as highVolume, lowVolume and mediumVolume) of the table to make informed decisions.

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 of 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
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:
  ```sql
  alter system set encryption key authenticated by "keypasswd"
  ```

• The wallet can be closed or opened using the following commands:
  ```sql
  alter system set wallet open identified by "keypasswd";
  alter system set wallet close;
  ```

• Column level encryption can be achieved using the following commands:
  ```sql
  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:
  ```sql
  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 the Final Measurement table (D1_MSRMT), keep the current table partition uncompressed. All of the older partitions will be compressed based on QUERY HIGH compression.

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

• For the Initial Measurement Data table (D1_INIT_MSMRT_DATA), always keep CLOBs in securefiles and with MEDIUM compression. Also 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.
Examples:
Create a Tablespace with Advanced Rowstore Compress

```
CREATE BIGFILE TABLESPACE CM_XT012_P2017JAN DATAFILE '+DATA' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
```

Create Table with Subpartitions using Compressed Tablespaces & Securefiles Compression

```
CREATE TABLE CI_ADJ (
    ADJ_ID CHAR(12) NOT NULL ENABLE,
    SA_ID CHAR(10) DEFAULT ' ' NOT NULL ENABLE, ADJ_TYPE_CD CHAR(8) DEFAULT ' ' NOT NULL ENABLE, ADJ_STATUS_FLG CHAR(2) DEFAULT ' ' NOT NULL ENABLE, CRE_DT DATE,
    CAN_RSN_CD CHAR(4) DEFAULT ' ' NOT NULL ENABLE, ADJ_AMT NUMBER(15,2) DEFAULT 0 NOT NULL ENABLE, XFER_ADJ_ID CHAR(12) DEFAULT ' ' NOT NULL ENABLE, CURRENCY_CD CHAR(3) DEFAULT ' ' NOT NULL ENABLE, COMMENTS VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE, VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE, BEHALF_SA_ID CHAR(10) DEFAULT ' ' NOT NULL ENABLE, BASE_AMT NUMBER(15,2) DEFAULT 0 NOT NULL ENABLE, GEN_REF_DT DATE,
    APPR_REQ_ID CHAR(12) DEFAULT ' ' NOT NULL ENABLE, ADJ_DATA_AREA CLOB, ILM_DT DATE,
    ILM_ARCH_SW CHAR(1),)
ENABLE ROW MOVEMENT
PARTITION BY RANGE (ILM_DT)
SUBPARTITION BY RANGE (ADJ_ID) SUBPARTITION TEMPLATE (SUBPARTITION S01 VALUES LESS THAN ( '124999999999' ), SUBPARTITION S02 VALUES LESS THAN ( '249999999999' ), SUBPARTITION S03 VALUES LESS THAN ( '374999999999' ), SUBPARTITION S04 VALUES LESS THAN ( '499999999999' ), SUBPARTITION S05 VALUES LESS THAN ( '624999999999' ), SUBPARTITION S06 VALUES LESS THAN ( '749999999999' ), SUBPARTITION S07 VALUES LESS THAN ( '874999999999' ), SUBPARTITION S08 VALUES LESS THAN ( MAXVALUE )
)
PARTITION "P2017JAN" VALUES LESS THAN (TO_DATE('2017-02-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017JAN,
PARTITION "P2017FEB" VALUES LESS THAN (TO_DATE('2017-03-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017FEB,
PARTITION "P2017MAR" VALUES LESS THAN (TO_DATE('2017-04-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
```
tablespace CM_XT012_P2017MAR,
PARTITION "P2017APR" VALUES LESS THAN (TO_DATE('2017-05-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017APR,
PARTITION "P2017MAY" VALUES LESS THAN (TO_DATE('2017-06-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017MAY,
PARTITION "P2017JUN" VALUES LESS THAN (TO_DATE('2017-07-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017JUN,
PARTITION "P2017JUL" VALUES LESS THAN (TO_DATE('2017-08-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017JUL,
PARTITION "P2017AUG" VALUES LESS THAN (TO_DATE('2017-09-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017AUG,
PARTITION "P2017SEP" VALUES LESS THAN (TO_DATE('2017-10-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017SEP,
PARTITION "P2017OCT" VALUES LESS THAN (TO_DATE('2017-11-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017OCT,
PARTITION "P2017NOV" VALUES LESS THAN (TO_DATE('2017-12-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017NOV,
PARTITION "P2017DEC" VALUES LESS THAN (TO_DATE('2017-01-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017DEC,
PARTITION "PMAX" VALUES LESS THAN (MAXVALUE)
tablespace CM_XT012_PMAX
);
Create a Compressed Local Index

CREATE UNIQUE INDEX XT012S3 ON CI_ADJ ( ILM_DT, ILM_ARCH_SW, ADJ_ID ) TABLESPACE CM_XT012_IND COMPRESS ADVANCED LOW;

Create a Compressed Global Partitioned Index

CREATE UNIQUE INDEX XT012S2 ON CI_ADJ ( XFER_ADJ_ID, ADJ_ID ) TABLESPACE CM_XT012_IND
GLOBAL PARTITION BY HASH (XFER_ADJ_ID, ADJ_ID) (
PARTITION PART1 TABLESPACE CM_XT012_IND, PARTITION PART2 TABLESPACE CM_XT012_IND, PARTITION PART3 TABLESPACE CM_XT012_IND, PARTITION PART4 TABLESPACE CM_XT012_IND, PARTITION PART5 TABLESPACE CM_XT012_IND, PARTITION PART6 TABLESPACE CM_XT012_IND, PARTITION PART7 TABLESPACE CM_XT012_IND, PARTITION PART8 TABLESPACE CM_XT012_IND
)
COMPRESS ADVANCED LOW;
Do NOT specify standard index compression.

CREATE INDEX XT012S1 ON CI_ADJ ( SA_ID, ADJ_TYPE_CD ) TABLESPACE CM_XT012_IND LOCAL COMPRESS 1 COMPRESS ADVANCED LOW;

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)');
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 MDM and Information Lifecycle Management and Data Archiving in CCB 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)
    CONSTRAINT <> 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 works 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 Statistic

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;
/

Materialized View


Prerequisites
Please make sure the following:

1. Set parameter QUERY_REWRITE_ENABLED=TRUE at database level.
   Use the following SQL:
   
   ```sql
   ALTER SYSTEM SET QUERY_REWRITE_ENABLED=TRUE; OR
   ALTER SYSTEM SET QUERY_REWRITE_ENABLED=TRUE SCOPE=BOTH;
   ```

2. To create a materialized view in another user's schema you must have the CREATE ANY MATERIALIZED VIEW system privilege. The owner of the materialized view must have the CREATE TABLE system privilege. The owner must also have access to any master tables of the materialized view that the schema owner does not own (for example, if the master tables are on a remote database) and to any materialized view logs defined on those master tables, either through a SELECT object privilege on each of the tables or through the SELECT ANY TABLE system privilege.

3. To create a refresh-on-commit materialized view (ON COMMIT REFRESH clause), in addition to the preceding privileges, you must have the ON COMMIT REFRESH object privilege on any master tables that you do not own or you must have the ON COMMIT REFRESH system privilege.

To create the materialized view with query rewrite enabled, in addition to the preceding privileges: If the schema owner does not own the master tables, then the schema owner must have the GLOBAL QUERY REWRITE privilege or the QUERY REWRITE object privilege on each table outside the schema.

```sql
CREATE MATERIALIZED VIEW F1_BO_LIFECYCLE_STATUS_MVW
(
BUS_OBJ_CD,
LIFE_CYCLE_BO_CD,
BO_STATUS_CD,
BATCH_CD
)
BUILD IMMEDIATE REFRESH ON COMMIT ENABLE QUERY REWRITE AS
SELECT
BO2.BUS_OBJ_CD,BO.LIFE_CYCLE_BO_CD,BOSA.BO_STATUS_CD,LCBOS.BATCH_CD as LC_BATCH_CD
FROM
F1_BUS_OBJ BO2,
F1_BUS_OBJ BO,
F1_BUS_OBJ_STATUS LCBOS,
F1_BUS_OBJ_STATUS_ALG BOSA
WHERE
BO2.LIFE_CYCLE_BO_CD =BO.LIFE_CYCLE_BO_CD AND
BO.BUS_OBJ_CD = BOSA.BUS_OBJ_CD AND
BOSA.BO_STATUS_SEVT_FLG = 'F1AT' AND
LCBOS.BUS_OBJ_CD = BO.LIFE_CYCLE_BO_CD AND
LCBOS.BO_STATUS_CD = BOSA.BO_STATUS_CD
```
/
create synonym SPLUSR.F1_BO_LIFECYCLE_STATUS_MVW for SPLADM.F1_BO_LIFECYCLE_STATUS_MVW;
grant select on F1_BO_LIFECYCLE_STATUS_MVW to FW_DEV;
grant select on F1_BO_LIFECYCLE_STATUS_MVW to SPL_USER;
grant select on F1_BO_LIFECYCLE_STATUS_MVW to SPL_READ;

For more information, refer to the following documents:

- Basic Query Rewrite (Oracle 11g) - https://docs.oracle.com/cd/B28359_01/server.111/b28313/qrbasic.htm
- Basic Query Rewrite for Materialized Views (Oracle 12c) - https://docs.oracle.com/database/121/DWHSG/qrbasic.htm#DWHSG01813
- Troubleshooting Materialized Views - http://docs.oracle.com/database/121/ARPLS/d_mview.htm#ARPLS67193
- Debugging materialized Views - http://docs.oracle.com/cd/B28359_01/server.111/b28313/qradv.htm

**Known Issues**
The following are some of the known issues at the time of release. For more information, refer to these articles on My Oracle Support:

- Query Did Not Rewrite For A User Other Than The Owner Of the Materialized View (Doc ID 1594725.1) - A patch is available for bug report 14772096 for some platforms.
- Query rewrite not working as expected with SELECT DISTINCT (Doc ID 766113.8) for Oracle version – 11.2.0.1 and 11.1.0.7 Fixed in version - 12.1.0.1 (Base Release), 11.2.0.2 (Server Patch Set)
Chapter 5
Conversion Tools

This section describes the following database conversion tools:

- Database Configuration
- Script Installation
- Preparing the Production Database
- Preparing the Staging Database

Note: All database related single fixes and service packs need to be applied against the production schema. Staging schema should not be updated with database single fixes or service packs. Staging schema need to be rebuilt for any fixes that contain DDL to create new database objects in production schema.
Database Configuration

The Conversion Tool Kit requires at least two sets of schema. One is to hold the staging data that the conversion tool gets the data from and performs validations. We call this schema the staging database. The target schema, which is referred to as the production database, is where the conversion tool inserts the validated data. Both the production database and the staging databases can reside in a single Oracle database or in different databases that are connected via a database link. Only the single database configuration is supported.

The following schematic diagram shows a sample configuration of both the production and staging environments in which the Conversion Tool Kit operates. The production and staging databases must be the same release level.

All the tables and views for the application are defined in the production database. The staging database has the same set of tables and views as the production database, except the tables that are grouped as part of the business configuration (control tables). Details on the differences of the tables of the two databases and of the conversion tool functionality are found in the Conversion Tool document.

Script Installation

The Conversion Setup Utility, ConvSetup.exe, is provided in this release of Oracle Utilities Customer To Meter to set up conversion schemas.

Install the Oracle Client 12c on Windows desktop and configure SQLNet to connect to the target database. The Conversion folder contains the conversion setup utility: ConvSetup.exe and Conversion.bat.
This section of the document describes how to create the databases for the conversion tool kit.

**Preparing the Production Database**

If the production database does not exist create the database under the production schema owner (CISADM).

If the production database is upgraded from the previous version of the application make sure all public synonyms that are created on the application tables are deleted. Instead, each application user should have private synonyms created on the application tables in order for the conversion tool configuration to work.

**Preparing the Staging Database**

Once you have created a staging owner (STGADM), application user (STGUSER) and read access user (STGREAD), install the initial database option in the staging schema. The rest of the steps are listed below.

Run ConvSetup.exe from under the Conversion folder. The script prompts you for the following values:

- Database Platform: Oracle (O)
- Database connection information
- Database Name
- System Password
- Production Schema Name
- Staging Schema Name
- Read-Write user for Staging Schema.

ConvSetup.exe performs following tasks:

- Creates cx* views on the master/transaction tables in the production database.
- Grants the privileges on the master/transaction tables in the production database to the staging owner.
- Drops control tables and creates views on production control tables in the staging database.
- Grants privileges on the control tables to the staging owner.
- Grants privileges on the cx* views to the staging application user.
- Creates generated key tables.
- Creates generated table primary key and secondary indexes.

In addition to above tasks ConvSetup.exe also generates the following SQL scripts:

- create_cxviews.sql
- create_ctlviews.sql
• createck_tbls.sql
• create_grants.sql
• createck_pkix.sql
• createck_secix.sql

By default the conversion.bat updates all changes to the staging schema. If you want to generate only the above sql scripts and not apply changes to staging schema then update conversion.bat by removing “-u”. The sql scripts can be applied to the staging schema later. The sqls scripts need to be executed in the same order as described above using SQL*PLUs.

Once the staging schema has been set up, generate the security for the staging user using:

oragensec.exe -d stgadm, MCPASSWORD, DATABASE_NAME -r
stg_read, stg_user -u stguser
Oracle Utilities Customer To Meter provides support for Information Lifecycle Management (ILM) and Data Archiving.

ILM is a process 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 passes the data is accessed less frequently or not at all. The costs of storing data that are accessed infrequently can be reduced by moving the data to lower cost mass storage media. Typically this involves a trade-off between cost and increased access times. Based on business needs, data may eventually be archived and purged from the database and kept offline ready to be restored if required.

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 application and database configuration and requires Oracle Partitioning.

An underlying design principle of the Oracle Utilities Application Framework ILM implementation is the concept that the age of the data may not be the only criterion used to determine when a record is able to be archived. There may be business rules that dictate that some records are still current and must not be archived yet.

ILM enabled objects have a combination of an ILM date and an ILM Archive Switch. The ILM date is used in conjunction with partitioning to group data by age. The ILM Archive Switch is set by a background process when the record meets the business rules specific to that Maintenance Object if the record is eligible to be archived. The ILM Archive Switch gives Database Administrators an easy method to check when all records in a partition meet the business criteria that make the partition eligible to be archived. If the ILM Archive Switch is set for all records, then the DBA can take the steps required to archive the partition.

Moving data between storage tiers takes advantage of the partitioning by ILM Date but does not require that the ILM Archive Switch is set. Oracle recommends using the Oracle Database ILM Assistant to assist with this process.

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 date column is defaulted to an appropriate date (typically the system date) when a new record is inserted, the MO is partitioned on the ILM_DT, so it should only be updated in exceptional circumstances as this would cause the record to be deleted from its current partition and inserted into a different partition, which is a relatively expensive operation.
  - 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 indicating that the record is eligible to be archived.
- Database Referential Integrity Constraints - These are required for reference partitioning of Child tables of ILM enabled MOs
- Partitioning - Partitioning is mandatory for ILM implementation. It is used to separate the data by ILM date so that data of a similar age is kept together.
- One Tablespace per Partition - The ILM implementation requires that each MO partition resides in a dedicated tablespace so that they can be easily managed.
- Naming Convention - This section covers the recommended naming convention to be used for partitions/subpartitions and tablespaces.
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.

- **Business FlagOn-going Maintenance Phase** - This phase covers the ongoing maintenance tasks such as add partition, archive and restore partitions.

**Preparation Phase**

**Note**: In order to successfully implement ILM as described here, the following DB Version and Patch are pre-requisites: version 12.1.0.2.0 and Patch 15996848.

The steps needed to enable ILM functionality differ depending on whether ILM is enabled as part of the initial implementation of the product or enabled ILM on an existing implementation where data already exists in the respective tables.

- **Initial Install** – For an initial installation, the section Module Specific ILM Implementation Details outlines the additional steps to be performed on base delivered ILM Enabled Tables to conform to ILM requirements. In addition, Appendix A: Sample SQL for Enabling ILM in CCB (Initial Install) provides sample reference DDLs using two maintenance objects as examples.

- **Transform NON-ILM implementation to ILM Enabled Implementation**:
  The following steps provide a high level overview of steps that must be performed to implement ILM on enabled MOs for an existing implementation. Please refer to Appendix B: Sample SQL For Enabling ILM in CCB (Existing Installation) for detailed information using To Do Entry as an example.

1. Rename the existing tables (Parent table followed by child table(s)), 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 Key index.
7. Create Primary Key Constraint 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 B: Sample SQL For Enabling ILM in CCB (Existing Installation). Please note that child table should be partitioned using reference partitioning of the parent table’s partitioning key.

10. Drop the original, renamed tables after verifying the newly created partitioned tables.

---

**On-going Maintenance Phase**

The following steps provide a high level overview of what needs to be done for on-going maintenance for ILM on enabled MOs.

Please refer to the Appendix C: Sample SQL for Periodic Maintenance for detailed information using two maintenance objects as examples.

1. Add the partition:
   a. Create Tablespace to be used for the new parent table partition.
   b. Since, we define MAXVALUE Partition; new partition can only be created using “SPLIT” operation. Identify and use next HIGH_VALUE Partition for the split operation.
   c. All the child table(s) partition(s)/LOB(s) must be altered to use the same tablespace as that of the parent table’s partition.
   d. Enable advanced compression on all child table(s).
   e. Copy partition level statistics from the previous partition

2. Archive the partition:
   a. Make the tablespace that will be archived READ ONLY.
   b. Check that no records have ILM_ARCH_SW = ‘N’.
      - If record count is zero, proceed with further steps.
      - If record count is not zero, then change the tablespace back to READ WRITE MODE as Archive is not Feasible at the time.
   c. Create an archive tablespace for the partition that needs to be archived.
   d. Create staging tables using the new archive tablespace. Load data for all child tables first.
   e. Create staging table using the new archive tablespace and load data for the parent table.
   f. Export tablespace using TRANSPORT_TABLESPACES method.
      Make Sure Tablespace datafile required for further import is preserved.
   g. Drop the partition, partition the tablespace and archive the tablespace (as it is already exported).
3. Restore the partition:
   a. Create a new tablespace to restore the partition.
   b. Add partition using split operation on next greater high value partition.

   If the table contains LOBS, there will an additional statement in split partition DDL indicating tablespace where the LOBs will be stored.
   c. Enable advanced compression on all child table(s).
   d. Import Tablespace using TRANSPORT_TABLESPACES method.
   e. Load data into the parent table first from the staging table
   f. Load data into the child table from the staging table
   g. Drop the archive tablespace after import and data loading is successful.

4. Move Data between different storage tiers:

The ILM facilities can be used within the database to implement storage savings, as follows:

   • 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.
   • 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.

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 for when data is moved it can be automatically compressed to desired degree.
   • 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

### Naming Convention

The naming convention for tablespace, partitions & subpartition is standardized as follows:

- Each name consists of some or all of the following parts
- The parts of the name are organized hierarchically
- Each part of the Name is separated with an underscore.
- The maximum name length must not exceed 30 Characters.
- For an MO, the parent table and child table share the same tablespace for the corresponding partition (or sub partition as appropriate).
- Square brackets [ ] indicate that this part of the name should be omitted if not required.

**OWNERFLAG_TABLEIDENTIFIER_PARTITIONNAME[_SUBPARTITIONNAME][_ARCHIVEFLAG][_COMPRESSFLAG]**

For details on the convention, please refer to the table below:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNERFLAG</td>
<td>Owner flag for the relevant application for example “C1” for CCB</td>
</tr>
<tr>
<td>Convention</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TABLE IDENTIFIER</td>
<td>The Index Name of the Primary Key index without the “P0” suffix. For example, if the PK index name is XT039P0, the table identifier would be “XT039”.</td>
</tr>
<tr>
<td>PARTITION NAME</td>
<td>The Partition name should be prefixed with a P followed by a name which conforms to one of the following standards:</td>
</tr>
<tr>
<td></td>
<td>• 4 digit year and 3 letter month abbreviation PYYYYMON corresponding to the ILM date e.g. P2017JAN</td>
</tr>
<tr>
<td></td>
<td>• PMAX if it is the Max Value partition</td>
</tr>
<tr>
<td>SUBPARTITION NAME</td>
<td>If subpartitions are used, name should be prefixed with S followed by a name of not more than 5 characters which conforms to the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• SMAX if this is the Max Value sub partition</td>
</tr>
<tr>
<td></td>
<td>• If the sub partition holds data for a sub retention period use a number equal to that period e.g S91 if the sub retention period &lt; 91 days.</td>
</tr>
<tr>
<td></td>
<td>• For a range based SubPartition on Primary Key, use an integral number increasing by +1. For example, if there are 8 sub partitions use S01 through S08</td>
</tr>
<tr>
<td>ARCHIVEFLAG</td>
<td>This flag is used as a suffix to the table and tablespace name for the staging tables created for the archiving operation.</td>
</tr>
<tr>
<td></td>
<td>• ARC</td>
</tr>
<tr>
<td>Convention</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>COMPRESS FLAG</td>
<td>This flag is used as a suffix to the tablespace name for the staging tables created when compressing a partition.</td>
</tr>
<tr>
<td></td>
<td>• C</td>
</tr>
<tr>
<td></td>
<td>For compression related tasks, this is used as suffix to the tablespace name.</td>
</tr>
<tr>
<td></td>
<td>• Partition Tablespace Name: It is formed by OWNERFLAG_TABLEIDENTIFIER_PAR TITIONNAME.</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>CM_D1T304_PMAX</td>
</tr>
<tr>
<td></td>
<td>CM_D1T304_P2017JAN</td>
</tr>
<tr>
<td></td>
<td>• SubPartition Tablespace Name: It is formed by OWNERFLAG_TABLEIDENTIFIER_PAR TITIONNAME_SUBPARTITIONNAME.</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>CM_D1T304_PMAX_SMAX</td>
</tr>
<tr>
<td></td>
<td>CM_D1T304_P2017JAN_SMAX</td>
</tr>
<tr>
<td></td>
<td>CM_D1T304_PMAX_S001</td>
</tr>
<tr>
<td></td>
<td>CM_D1T304_P2017JAN_S181</td>
</tr>
<tr>
<td></td>
<td>• Archive Staging Table And Its Tablespace Name (When archiving partition): It is formed by OWNERFLAG_TABLEIDENTIFIER_PAR TITIONNAME_ARCHIVEFLAG.</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>CM_D1T304_P2017JAN_ARC</td>
</tr>
<tr>
<td></td>
<td>• Archive Staging Table And Its Tablespace Name (When archiving subpartition): It is formed by OWNERFLAG_TABLEIDENTIFIER_PAR TITIONNAME_SUBPARTITIONNAME_ARCHIVEFLAG.</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>CM_D1T304_P2017JAN_S181_ARC</td>
</tr>
<tr>
<td></td>
<td>• Compressed Tablespace name (When compressing partition):</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>CM_D1T304_P2017JAN_C</td>
</tr>
</tbody>
</table>
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.

All indexes are listed with a recommendation whether the index should be global or local and whether the index should be partitioned. In addition to the base delivered indexes, each 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. The recommended column that should be used to populate the ILM_DT is also shown. Please refer to Appendix B: Sample SQL For Enabling ILM in CCB (Existing Installation) for sample DDL(s).

To Do Entry

This table describes the To Do Entry maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_TD_ENTRY (Parent)</td>
<td>RANGE (ILM_DT, TD_ENTRY_ID)</td>
<td>XT039P0</td>
<td>TD_ENTRY_ID</td>
<td>Global Partitioned</td>
<td>RANGE (TD_ENTRY_ID)</td>
<td>CI_TD_ENTRY. CRE_DTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT039S2</td>
<td>ASSIGNED_TO, TD_ENTRY_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT039S3</td>
<td>ENTRY_STATUS_FLG, ASSIGNED_TO</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT039S4</td>
<td>ROLE_ID, TD_TYPE_CD, ENTRY_STATUS_FLG, TD_PRIORITY_FLG</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT039S5</td>
<td>BATCH_CD, BATCH_NBR, ENTRY_STATUS_FLG</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT039S6</td>
<td>TD_ENTRY_ID, ASSIGNED_TO, ENTRY_STATUS_FLG</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT039S7</td>
<td>COMPLETE_USER_ID, COMPLETE_DTTM, TD_ENTRY_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_</td>
<td>ILM_DT, ILM_ARCH_SW, TD_ENTRY_ID</td>
<td>XT039S8</td>
<td></td>
<td>Local Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_TD_ENTRY_</td>
<td>Reference Partitioning</td>
<td>XT701P0</td>
<td>TD_ENTRY_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHA</td>
<td></td>
<td>XT701S1</td>
<td>SRCH_CHAR_VAL, CHAR_TYPE_CD, TD_ENTRY_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT701S2</td>
<td>CHAR_VAL_FK1</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Sync Request (Outbound)
This table describes the Sync Request (Outbound) maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_SYNC_REQ</td>
<td>RANGE (ILM_DT, F1_SYNC_REQ_ID)</td>
<td>F1T014P0</td>
<td>F1_SYNC_REQ_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>F1_SYNC_REQ.CRE_DTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T014S1</td>
<td>BO_STATUS_CD, BUS_OBJ_CD, F1_SYNC_REQ_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T014S2</td>
<td>BO_STATUS_REASON_CD</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table Name</td>
<td>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</td>
<td>Index Name</td>
<td>Index Columns</td>
<td>Index Type</td>
<td>ILM_DT Initial Load</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------</td>
<td>-------------</td>
<td>---------------</td>
<td>------------</td>
<td>---------------------</td>
<td></td>
</tr>
<tr>
<td>FIT014S3</td>
<td>MAINT_OBJ_CD, PK_VALUE1, PK_VALUE2, F1_SYNC_REQ_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_FIT014S4</td>
<td>ILM_DT, ILM_ARC_SW, F1_SYNC_REQ_ID</td>
<td>Local Partitioned</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_SYNC_REQ_CCHAR</td>
<td>Reference Partitioning</td>
<td>FIT017P0</td>
<td>F1_SYNC_REQ_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIT017S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_SYNC_REQ_EXTRACT</td>
<td>Reference Partitioning</td>
<td>FIT019P0</td>
<td>F1_SYNC_REQ_ID, SEQ_NUM</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_SYNC_REQ_L_OG</td>
<td>Reference Partitioning</td>
<td>FIT015P0</td>
<td>F1_SYNC_REQ_ID, SEQNO</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIT015S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIT015S2</td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIT015S3</td>
<td>BO_STATUS_REASON</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_SYNC_REQ_L_OG_PARM (Child Table of F1_SYNC_REQ_L_OG_PARM)</td>
<td>Reference Partitioning</td>
<td>FIT016P0</td>
<td>F1_SYNC_REQ_ID, SEQNO, PARM_SEQ</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** It is recommended that data retention policies and rules for this object match the policies and rules implemented for the Inbound Sync Request on the target system to avoid data inconsistencies when auditing.
Inbound Sync Request
This table describes the Inbound Sync Request maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_SYNC_REQ_IN</td>
<td>RANGE/ILM_DT, F1_SYNC_REQ_IN_ID</td>
<td>F1T191P0</td>
<td>F1_SYNC_REQ_IN_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>F1_SYNC_REQ_IN_ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T191S1</td>
<td>BO_STATUS_CD, BUS_OBJ_CD, F1_SYNC_REQ_IN_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T191S2</td>
<td>MAINT_OBJ_CD, EXT_PK_VALUE, NT_XID_CD, PK_VALUE</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T191S3</td>
<td>EXT_REFERERENCE_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_F1T191S3</td>
<td>ILM_DT, ILM_ARCH_SW, F1_SYNC_REQ_IN_ID</td>
<td>F1T193P0</td>
<td>F1_SYNC_REQ_IN_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T193S1</td>
<td>SRCH_CHAR_VALUE</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_SYNC_REQ_IN_CHAR</td>
<td>Reference Partitioning</td>
<td>F1T197P0</td>
<td>F1_SYNC_REQ_IN_ID, SEQNO</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>F1_SYNC_REQ_IN_EXCP</td>
<td>Reference Partitioning</td>
<td>F1T198P0</td>
<td>F1_SYNC_REQ_IN_ID, SEQNO, PARM_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>F1_SYNC_REQ_IN_LOG</td>
<td>Reference Partitioning</td>
<td>F1T194P0</td>
<td>F1_SYNC_REQ_IN_ID, SEQNO</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T194S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T194S2</td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Note: It is recommended that data retention policies and rules for this object match the policies and rules implemented for the Outbound Sync Request on the source system to avoid data inconsistencies when auditing.

### Outbound Message

This table describes the Outbound Message maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_SYNC_REQ_IN _LOG_PARM (Child Table of F1_SYNC_REQ_IN _LOG)</td>
<td>Reference Partitioning</td>
<td>FIT195P0</td>
<td>F1_SYNC_REQ_IN_I N_ID, SEQNO, PARM_SEQ</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_SYNC_REQ_IN _REI_OBJ</td>
<td>Reference Partitioning</td>
<td>FIT192P0</td>
<td>F1_SYNC_REQ_I N_ID, MAINT_OBJ_CD, REI_OBJ_TYPE_, FLG</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FIT192S1</td>
<td>PK_VALUE1</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_OUTMSG (Parent)</td>
<td>RANGE (ILM_DT, OUTMSG_ID)</td>
<td>FT010P0</td>
<td>OUTMSG_ID</td>
<td>Global Partitioned</td>
<td>F1_OUTMSG. CRE_DTTM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT010S1</td>
<td>OUTMSG_STAT US_FLG, OUTMSG_TYPE_CD</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_FTO10S2</td>
<td></td>
<td></td>
<td>ILM_DT, ILM_ARC_SW, OUTMSG_ID</td>
<td>Local Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_OUTMSG_ ERRPARAM</td>
<td>Reference Partitioning</td>
<td>FT011P0</td>
<td>OUTMSG_ID, PARM_SEQ</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Service Task

This table describes the Service Task maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_SVC_TASK (Parent)</td>
<td>RANGE (ILM_DT, F1_SVC_TASK_ID)</td>
<td>F1C474P0</td>
<td>F1_SVC_TASK_ID</td>
<td>Global</td>
<td>Partitioned</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C474S1</td>
<td>F1_STASK_TYPE_CD</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C474S2</td>
<td>BUS_OBJ_CD</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CM_ILM_F1C474S2</td>
<td>ILM_DT, ILM_ARC_SW, F1_SVC_TASK_ID</td>
<td>Local</td>
<td>Partitioned</td>
</tr>
<tr>
<td>F1_SVC_TASK_CHAR</td>
<td>Reference Partitioning</td>
<td>F1C476P0</td>
<td>F1_SVC_TASK_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C476S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td>F1_SVC_TASK_LOG</td>
<td>Reference Partitioning</td>
<td>F1C477P0</td>
<td>F1_SVC_TASK_ID, SEQNO</td>
<td>Global</td>
<td>Partitioned</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C477S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C477S2</td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td>F1_SVC_TASK_LOG_PARM (Child Table of F1_SVC_TASK_LOG)</td>
<td>Reference Partitioning</td>
<td>F1C478P0</td>
<td>F1_SVC_TASK_ID, SEQNO, PARM_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
</tr>
<tr>
<td>F1_SVC_TASK_REL_OBJ</td>
<td>Reference Partitioning</td>
<td>F1C479P0</td>
<td>F1_SVC_TASK_ID, MAINT_OBJ_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
</tr>
</tbody>
</table>
### Object Revision

This table describes the Object Revision maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Local or Global</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_OBJ_REV (Parent)</td>
<td>RANGE (ILM_DT, REV_ID)</td>
<td>MAINT_OBJ_CD, PK_VALUE1</td>
<td>RANGE (REV_ID)</td>
<td>F1_OBJ_REV. STATUS_UPD_D TTM</td>
<td>Partitioned</td>
<td>ILM_DT</td>
</tr>
<tr>
<td>FT035P0</td>
<td>REV_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>FT035S1</td>
<td>BO_STATUS_CD, BUS_OBJ_CD, REV_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FT035S2</td>
<td>MAINT_OBJ_CD, PK_VALUE1</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FT035S3</td>
<td>EXT_REFERENCE_ID, MAINT_OBJ_CD</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FT035S4</td>
<td>USER_ID, MAINT_OBJ_CD</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FT035S5</td>
<td>PK_VALUE1</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_FT035S 6</td>
<td>ILM_DT, ILM_ARC_SW, REV_ID</td>
<td>Local</td>
<td></td>
<td></td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>Table Name</td>
<td>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</td>
<td>Index Name</td>
<td>Index Columns</td>
<td>Index Type Global or Local</td>
<td>Index Partitioning Sub-Partitioning Key</td>
<td>ILM_DT Initial Load</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------</td>
<td>------------</td>
<td>--------------------------------</td>
<td>---------------------------</td>
<td>----------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>F1_OBJ_REV_CHAR</td>
<td>Reference Partitioning</td>
<td>FT037P0</td>
<td>REV_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT037S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_OBJ_REV_LOG</td>
<td>Reference Partitioning</td>
<td>FT039P0</td>
<td>REV_ID, SEQNO</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Child Table of F1_OBJ_REV_LOG)</td>
<td>FT040P0</td>
<td>REV_ID, SEQNO, PARM_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
</tbody>
</table>

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.

### Adjustment

This table describes the Adjustment maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_ADJ (Parent)</td>
<td>RANGE (ILM_DT, ADJ_ID)</td>
<td>XT012P0</td>
<td>ADJ_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_ADJ (Parent)</td>
<td>RANGE (ILM_DT, ADJ_ID)</td>
<td>XT012P0</td>
<td>ADJ_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_ADJ (Parent)</td>
<td>RANGE (ILM_DT, ADJ_ID)</td>
<td>XT012P0</td>
<td>ADJ_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_ADJ (Parent)</td>
<td>RANGE (ILM_DT, ADJ_ID)</td>
<td>XT012P0</td>
<td>ADJ_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
</tbody>
</table>

Information Lifecycle Management and CC&B Data Archiving in C2M 6 - 16
Oracle Utilities Customer To Meter Database Administrator’s Guide
### Approval Request

This table describes the Approval Request maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CL_APPR_REQ (Parent)</td>
<td>RANGE (ILM_DT, APPR_REQ_ID)</td>
<td>XT600P0</td>
<td>APPR_REQ_ID</td>
<td>Global Partitioned</td>
<td>RANGE (APPR_REQ_ID)</td>
<td>MIN(LOG_DTT M) on CL_APPR_REQ_LOG for given APPR_REQ_ID</td>
</tr>
</tbody>
</table>

**Table 1:**

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_ADJ_APPR_REQ</td>
<td>Reference Partitioning</td>
<td>XT160P0</td>
<td>AP_REQ_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT160S1</td>
<td>ADJ_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT160S2</td>
<td>BATCH_CD, BATCH_NBR</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_ADJ_CALC_LN</td>
<td>Reference Partitioning</td>
<td>XT310P0</td>
<td>ADJ_ID, SEQNO</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_ADJ_CL_CHAR</td>
<td>Reference Partitioning</td>
<td>XT309P0</td>
<td>ADJ_ID, SEQNO, CHAR_TYPE_CD</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_ADJ_CHAR</td>
<td>Reference Partitioning</td>
<td>XC781P0</td>
<td>ADJ_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XC781S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table Name</td>
<td>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</td>
<td>Index Name</td>
<td>Index Columns</td>
<td>Index Type Global or Local</td>
<td>Index Partitioning Key</td>
<td>ILM_DT Initial Load</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------</td>
<td>------------</td>
<td>----------------------------------------</td>
<td>----------------------------</td>
<td>------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>CI_BILL (Parent)</td>
<td>RANGE (ILM_DT, BILL_ID)</td>
<td>XT033P0</td>
<td>BILL_ID</td>
<td>Global</td>
<td>BILL_ID</td>
<td>ILM_DT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT033S1</td>
<td>ACCT_ID, BILL_STAT_FLG, BILL_CYC_CD, WIN_START_DT, CR_NOTE_FR_BILL_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT033S2</td>
<td>CR_NOTE_FR_BILL_ID, BILL_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bill

This table describes the Bill maintenance object.
<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>XT033S3</td>
<td></td>
<td>ALT_BILL_ID, BILL_STAT_FLG, BILL_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT033S4</td>
<td></td>
<td>OFFCYC_BGEN_ID, BILL_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT033S5</td>
<td></td>
<td>DOC_ID, DOC_TYPE_FLG, BILL_STAT_FLG, BILL_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_XT033S6</td>
<td>ILM_DT, ILM_ARCH_SW, BILL_ID</td>
<td>Local</td>
<td>Partitioned</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT033S7</td>
<td></td>
<td>ACCT_ID, OFFCYC_BGEN_ID, CRE_DTTM</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT033S8</td>
<td></td>
<td>LATE_PAY_CHARGE_SW, LATE_PAY_CHARGE_DT, BILL_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_BILL_CHAR</td>
<td>Reference Partitioning</td>
<td>XT313P0</td>
<td>BILL_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_BILL_EXCP</td>
<td>Reference Partitioning</td>
<td>XT038P0</td>
<td>BILL_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_BILL_MSGS</td>
<td>Reference Partitioning</td>
<td>XT091P0</td>
<td>BILL_ID, BILL_MSG_CD</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_BILL_MSG_PRM</td>
<td>Reference Partitioning</td>
<td>XT085P0</td>
<td>BILL_ID, BILL_MSG_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_BILL_SA</td>
<td>Reference Partitioning</td>
<td>XT046P0</td>
<td>BILL_ID, SA_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT046S1</td>
<td>SA_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Bill Segment
This table describes the Bill Segment maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_BILL_ROUTING</td>
<td>Reference Partitioning</td>
<td>XT075P0</td>
<td>BILL_ID, SEQNO</td>
<td>Global Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT075S1</td>
<td>BATCH_CD, BATCH_NBR, BILL_ID, NO_BATCH_PRT_SW</td>
<td>Global</td>
<td></td>
</tr>
</tbody>
</table>

### CI_BSEG (Parent) RANGE (ILM_DT, BSEG_ID)

<table>
<thead>
<tr>
<th>Index Name</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>XT048P0</td>
<td>BSEG_ID</td>
<td>Global Partitioned</td>
<td></td>
</tr>
<tr>
<td>XT048S1</td>
<td>BILL_ID</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td>XT048S2</td>
<td>SA_ID</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td>XT048S3</td>
<td>QUOTE_DTL_ID, BSEG_ID</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td>CM_ILM_XT048S4</td>
<td>ILM_DT, ILM_ARCH_SW, BSEG_ID</td>
<td>Local Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_BSEG_CALC</td>
<td>Reference partitioning</td>
<td>XT072P0</td>
<td>BSEG_ID, HEADER_SEQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT072S1</td>
<td>BILLABLE_CHG_ID, BSEG_ID</td>
</tr>
<tr>
<td>Table Name</td>
<td>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</td>
<td>Index Name</td>
<td>Index Columns</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------</td>
<td>------------</td>
<td>---------------</td>
</tr>
<tr>
<td>CI_BSEG_CALC_LN</td>
<td>Reference partitioning</td>
<td>XT050P0</td>
<td>BSEG_ID, HEADER_SEQ, SEQNO</td>
</tr>
<tr>
<td>CI_BSEG_CL_CHAR</td>
<td>Reference partitioning</td>
<td>XT056P0</td>
<td>BSEG_ID, HEADER_SEQ, SEQNO, CHAR_TYPE_CD</td>
</tr>
<tr>
<td>CI_BSEG_EXCP</td>
<td>Reference partitioning</td>
<td>XT051P0</td>
<td>BSEG_ID</td>
</tr>
<tr>
<td>CI_BSEG_MSG</td>
<td>Reference partitioning</td>
<td>XT080P0</td>
<td>BSEG_ID, BILL_MSG_CD</td>
</tr>
<tr>
<td>CI_BSEG_READ</td>
<td>Reference partitioning</td>
<td>XT054P0</td>
<td>BSEG_ID, SP_ID, SEQNO</td>
</tr>
<tr>
<td>CI_BSEG_SQ</td>
<td>Reference partitioning</td>
<td>XT055P0</td>
<td>BSEG_ID, UOM_CD, TOU_CD, SQL_CD</td>
</tr>
<tr>
<td>CI_BSEG_ITEM</td>
<td>Reference partitioning</td>
<td>XT053P0</td>
<td>BSEG_ID, SEQNO</td>
</tr>
</tbody>
</table>

XT054S1 SP_ID Global

XT054S2 START_REG_RE AD_ID Global

XT054S3 END_REG_READ_ID Global
### Statement
This table describes the Statement maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type</th>
<th>Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_STM (Parent)</td>
<td>RANGE (ILM_DT, STM_ID)</td>
<td>XT088P0</td>
<td>STM_ID</td>
<td>Global</td>
<td>STM_ID</td>
<td>CI_STM.STM_DT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CM_ILM_XT088S1</td>
<td>ILM_DT, ILM_ARCH_SW, STM_ID</td>
<td>Local</td>
<td>Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_STM_DTL</td>
<td></td>
<td>XT119P0</td>
<td>STM_DTL_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT119S1</td>
<td>STM_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Off Cycle Bill Generator
This table describes the Off Cycle Bill Generator maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type</th>
<th>Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1_OFFCYC_BGEN (Parent)</td>
<td>RANGE (ILM_DT, OFFCYC_BGEN_ID)</td>
<td>XT197P0</td>
<td>OFFCYC_BGEN_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>C1_OFFCYC_BGEN_STATUS_UP_D_DTTM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT197S1</td>
<td>ACCT_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CM_ILM_XT197S2</td>
<td>ILM_DT, ILM_ARCH_SW, OFFCYC_BGEN_ID</td>
<td>Local</td>
<td>Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1_OFFCYC_BGEN_ADJ</td>
<td>Reference partitioning</td>
<td>XT285P0</td>
<td>OFFCYC_BGEN_ID, ADJ_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Billable Charge
This table describes the Billable Charge maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>XT035S1</td>
<td>BILLABLE_CHG_ID</td>
<td>XT035P0</td>
<td>BILLABLE_CHG_ID</td>
<td></td>
<td>BILLABLE_CHG_ID</td>
</tr>
<tr>
<td>XT285S1</td>
<td>ADJ_ID</td>
<td>XT285P0</td>
<td>ADJ_ID</td>
<td>Global</td>
<td>ADJ_ID</td>
</tr>
<tr>
<td>CI_OFFCYC_BG_EN_BCHG</td>
<td>Reference partitioning</td>
<td>XT326S1</td>
<td>BILLABLE_CHG_ID, BILLABLE_CHG_ID</td>
<td>Partitioned</td>
<td>BILLABLE_CHG_ID</td>
</tr>
<tr>
<td>XT326S1</td>
<td>BILLABLE_CHG_ID</td>
<td>XT326P0</td>
<td>BILLABLE_CHG_ID</td>
<td>Global</td>
<td>BILLABLE_CHG_ID</td>
</tr>
<tr>
<td>CI_OFFCYC_BG_EN_CHAR</td>
<td>Reference partitioning</td>
<td>XT343S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td>SRCH_CHAR_VAL</td>
</tr>
<tr>
<td>XT343S1</td>
<td>SRCH_CHAR_VAL</td>
<td>XT343P0</td>
<td>SRCH_CHAR_VAL, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>SRCH_CHAR_VAL</td>
</tr>
<tr>
<td>CI_OFFCYC_BG_EN_LOG</td>
<td>Reference partitioning</td>
<td>XT344S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td>Global</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
</tr>
<tr>
<td>XT344S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td>XT344P0</td>
<td>CHAR_TYPE_CD, SEQNO</td>
<td>Global</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
</tr>
<tr>
<td>CI_OFFCYC_BG_EN_LOG_PARM</td>
<td>Reference partitioning</td>
<td>XT357S1</td>
<td>OFFCYC_BGEN_ID, SEQNO, PARM_SEQ</td>
<td>Global</td>
<td>OFFCYC_BGEN_ID, SEQNO, PARM_SEQ</td>
</tr>
<tr>
<td>CI_OFFCYC_BG_EN_SA</td>
<td>Reference partitioning</td>
<td>XT359S1</td>
<td>OFFCYC_BGEN_ID, SA_ID</td>
<td>Global</td>
<td>OFFCYC_BGEN_ID, SA_ID</td>
</tr>
<tr>
<td>XT359S1</td>
<td>SA_ID</td>
<td>XT359P0</td>
<td>SA_ID</td>
<td>Partitioned</td>
<td>SA_ID</td>
</tr>
<tr>
<td>CI_BILL_CHG (Parent)</td>
<td>RANGE (ILM_DT, BILLABLE_CHG_ID)</td>
<td>CI_BILL_CHG.S</td>
<td>RANGE (BILLABLE_CHG_ID)</td>
<td>TART_DT</td>
<td>CI_BILL_CHG.S TART_DT</td>
</tr>
</tbody>
</table>
### Case

This table describes the Case maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_CASE (Parent)</td>
<td>RANGE (ILM_DT, CASE_ID)</td>
<td>XT220P0</td>
<td>CASE_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT220S1</td>
<td>PER_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT220S2</td>
<td>ACCT_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Naming Convention

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>XT220S3</td>
<td></td>
<td>PREM_ID</td>
<td></td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT220S4</td>
<td></td>
<td>USER_ID</td>
<td></td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_XT220S</td>
<td>ILM_DT, ILM_ARCH_SW, CASE_ID</td>
<td></td>
<td></td>
<td>Local</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_CASE_CHAR</td>
<td>Reference Partitioning</td>
<td>XT222P0</td>
<td>CASE_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_CASE_LOG</td>
<td>Reference Partitioning</td>
<td>XT221P0</td>
<td>CASE_ID, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_CASE_LOG_PARM</td>
<td>Reference Partitioning</td>
<td>XT290P0</td>
<td>CASE_ID, SEQ_NUM, PARM_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
</tbody>
</table>

### Field Activity

This table describes the Field Activity maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_FA (Parent)</td>
<td>RANGE (ILM_DT, FA_ID)</td>
<td></td>
<td></td>
<td>RANGE (FA_ID)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT094P0</td>
<td>FA_ID</td>
<td></td>
<td></td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT094S1</td>
<td>SP_ID, FO_ID</td>
<td></td>
<td></td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT094S2</td>
<td>FO_ID, FA_ID</td>
<td></td>
<td></td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table Name</td>
<td>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</td>
<td>Index Name</td>
<td>Index Columns</td>
<td>Index Type Partitioning Key</td>
<td>Initial Load</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------------------------------</td>
<td>------------</td>
<td>---------------------------------------------</td>
<td>-----------------------------------</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>XT094S3</td>
<td></td>
<td>FA_STATUS_FLG, ELIG DISPATCH_SW, FO_ID, FA_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT094S4</td>
<td></td>
<td>APP_SCHED_ID, FA_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT094S5</td>
<td></td>
<td>TEST_SEL_ID, FA_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_XT094S6</td>
<td></td>
<td>ILM_DT, ILM_ARCH_SW, FA_ID</td>
<td>Local Partitioned</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT094S7</td>
<td></td>
<td>FA_EXT_ID, FA_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_FA_CHAR</td>
<td>Reference Partitioning</td>
<td>XT406P0</td>
<td>FA_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_FA_LOG</td>
<td>Reference Partitioning</td>
<td>XT350P0</td>
<td>FA_ID, SEQ_NUM</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_FA_REM</td>
<td>Reference Partitioning</td>
<td>XT407P0</td>
<td>FA_ID, FA_REM_CD</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_FA_REM_EXC</td>
<td>Reference Partitioning</td>
<td>XT312P0</td>
<td>FA_ID, FA_REM_CD</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_FA_REM_EXP</td>
<td>Reference Partitioning</td>
<td>XT405P0</td>
<td>FA_ID, FA_REM_CD, PARM_SEQ</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_FA_STEP</td>
<td>Reference Partitioning</td>
<td>XT095P0</td>
<td>FA_ID, STEP_SEQ_NBR</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT095S1</td>
<td></td>
<td>CC_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT095S2</td>
<td></td>
<td>SPAWNED_FA_ID, FA_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT095S3</td>
<td></td>
<td>MR_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Enrollment (Order)
This table describes the Enrollment (Order) maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_ENRL (Parent)</td>
<td>RANGE (ILI_D, ENRL_ID)</td>
<td>XT193P0</td>
<td>ENRL_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT193S1</td>
<td>PER_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT193S2</td>
<td>ACCT_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT193S3</td>
<td>PREM_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_XT193S4</td>
<td>ILM_DT, ILM_ARCH_SW, ENRL_ID</td>
<td>XT200P0</td>
<td>ENRL_ID, ENRL_ADDR_ENT_FLG</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT200S1</td>
<td>ADDRESS1_UPR, ENRL_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT200S2</td>
<td>CITY_UPR, ENRL_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_ENRL_ADDR</td>
<td>Reference Partitioning</td>
<td>XT191P0</td>
<td>ENRL_ID, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_ENRL_FLD</td>
<td>Reference Partitioning</td>
<td>XT199P0</td>
<td>ENRL_ID, ID_TYPE_CD</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_ENRL_LOG</td>
<td>Reference Partitioning</td>
<td>XT199S1</td>
<td>HASH_PER_ID_NBR, ENRL_ID, ID_TYPE_CD</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_ENRL_PER_ID</td>
<td>Reference Partitioning</td>
<td>XT194P0</td>
<td>ENRL_ID, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_ENRL_PER_NM</td>
<td>Reference Partitioning</td>
<td>XT194S1</td>
<td>ENTITY_NAME_UPR, ENRL_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_ENRL_PER_PHN</td>
<td>Reference Partitioning</td>
<td>XT195P0</td>
<td>ENRL_ID, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
</tbody>
</table>
### Payment Event
This table describes the Payment Event maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_PAY_EVENT (Parent)</td>
<td>RANGE (ILM_DT, PAY_EVENT_ID)</td>
<td>XT159P0</td>
<td>PAY_EVENT_ID</td>
<td>Global Partitioned</td>
<td>CI_PAY_EVENT, PAY_DT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT159S1</td>
<td>PAY_DT, PAY_EVENT_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT159S2</td>
<td>DOC_ID, PAY_EVENT_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CM_ILM_XT159S3</td>
<td>ILM_DT, ILM_ARCH_SW, PAY_EVENT_ID</td>
<td>Local Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_PAY_EVT_CHAR</td>
<td>Reference Partitioning</td>
<td>XT244P0</td>
<td>PAY_EVENT_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT244S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_PAY_EVT_EXCP</td>
<td>Reference Partitioning</td>
<td>XT161P0</td>
<td>PAY_EVENT_ID</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_PAY_TNDR</td>
<td>Reference Partitioning</td>
<td>XT265P0</td>
<td>PAY_TENDER_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT265S1</td>
<td>MICR_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT265S2</td>
<td>TENDER_AMT, PAY_TENDER_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table Name</td>
<td>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</td>
<td>Index Name</td>
<td>Index Columns</td>
<td>Index Type Global or Local</td>
<td>Index Partitioning Sub-Partitioning Key</td>
<td>ILM_DT Initial Load</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------------------------------</td>
<td>--------------</td>
<td>---------------------------------------------------</td>
<td>----------------------------</td>
<td>------------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>XT26S3</td>
<td></td>
<td>PAY_EVENT_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT26S4</td>
<td></td>
<td>PAYOR_ACCT_ID, TENDER_AMT</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT26S5</td>
<td></td>
<td>TNDR_CTL_ID, PAY_EVENT_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT26S6</td>
<td></td>
<td>ADJ_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT26S7</td>
<td></td>
<td>HASH_MICR_ID, PAY_TENDER_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_APAY_CLR_STG</td>
<td></td>
<td>XT003P0</td>
<td>APAY_CLR_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_PAY_TNDR_CHAR</td>
<td></td>
<td>XT003S1</td>
<td>BILL_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_PAY_TNDR_CHAR</td>
<td></td>
<td>XT003S2</td>
<td>PAY_TENDER_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_PAY_TNDR_CHAR</td>
<td></td>
<td>XT003S3</td>
<td>BATCH_CD, BATCH_NBR, SCHED_EXTRA CT_DT</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_PAY_TNDR_CHAR</td>
<td></td>
<td>XT003S4</td>
<td>ACCT_APAY_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_PAY_TNDR_CHAR</td>
<td></td>
<td>XT413P0</td>
<td>PAY_TENDER_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_PAY_TNDR_CHAR</td>
<td></td>
<td>XT413S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_PEVT_DST_DTL</td>
<td>Reference Partitioning</td>
<td>XT730P0</td>
<td>PAY_EVENT_ID, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_PEVT_DST_DTL</td>
<td></td>
<td>XT730S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Payment

This table describes the Payment maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_PAY (Parent)</td>
<td>RANGE (ILM_DT, PAY_ID)</td>
<td>XT156P0</td>
<td>PAY_ID</td>
<td>Global Partitioned</td>
<td>RANGE (PAY_ID)</td>
<td>CI_PAY_EVENT. PAY_DT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT156S1</td>
<td>ACCT_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT156S2</td>
<td>PAY_EVENT_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT156S3</td>
<td>PAY_AMT, PAY_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_XT156S</td>
<td>RANGE (ILM_DT, ILM_ARCH_SW, PAY_ID)</td>
<td>XT412P0</td>
<td>PAY_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_PAY_CHAR</td>
<td>Reference Partitioning</td>
<td>XT412S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_PAY_EXCP</td>
<td>Reference Partitioning</td>
<td>XT163P0</td>
<td>PAY_ID</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_PAY_SEG</td>
<td>Reference Partitioning</td>
<td>XT165P0</td>
<td>PAY_SEG_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT165S1</td>
<td>PAY_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT165S2</td>
<td>SA_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Match Event
This table describes the Match Event maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_MATCH_EVT (Parent)</td>
<td>RANGE (ILM_DT, MATCH_EVT_ID)</td>
<td>XT266P0</td>
<td>MATCH_EVT_ID</td>
<td>Global</td>
<td>RANGE (MATCH_EVT_ID)</td>
<td>CI_MATCH_EVT_CREATE_DT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT266S1</td>
<td>ACCT_ID, MEVT_STATUS_FLG</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT266S2</td>
<td>PAY_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_XT266S3</td>
<td>RANGE (ILM_DT, ILM_ARCH_SW, MATCH_EVT_ID)</td>
<td>CT368S2</td>
<td>BSEG_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CT368S3</td>
<td>USER_ID, USAGE_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Usage Request
This table describes the Usage Request maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1_USAGE (Parent)</td>
<td>RANGE (ILM_DT, USAGE_ID)</td>
<td>CT368S2</td>
<td>BSEG_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CT368S3</td>
<td>USER_ID, USAGE_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reference Partitioning</td>
<td></td>
<td>XT368P0</td>
<td>USAGE_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table Name</td>
<td>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</td>
<td>Index Name</td>
<td>Index Columns</td>
<td>Index Type Global or Local</td>
<td>Index Partitioning Sub-Partitioning Key</td>
<td>ILM_DT Initial Load</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------</td>
<td>-------------</td>
<td>---------------</td>
<td>---------------------------</td>
<td>----------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>XT368S1</td>
<td></td>
<td>BUS_OBJ_CD, BUS_OBJ_CD, BO_STATUS_CD, WIN_START_DT, BILL_CYC_CD, USAGE_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT368S2</td>
<td></td>
<td>BSEG_ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT368S3</td>
<td></td>
<td>USER_ID, USAGE_ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XT368S4</td>
<td></td>
<td>SA_ID</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT368S4</td>
<td></td>
<td>MASTER_USAG E_ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT368S5</td>
<td></td>
<td>SA_REL_ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_XT368S5</td>
<td></td>
<td>ILM_DT, ILM_ARCH_SW, USAGE_ID</td>
<td>Local</td>
<td>Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1_USAGE__ CHAR</td>
<td>Reference Partitioning</td>
<td>XT387P0</td>
<td>USAGE_ID, CHAR_TYPE_, CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>C1_USAGE__ LOG</td>
<td>Reference Partitioning</td>
<td>XT388P0</td>
<td>USAGE_ID, SEQNO</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>C1_USAGE__ LOG _PARAM</td>
<td>Reference Partitioning</td>
<td>XT389P0</td>
<td>USAGE_ID, SEQNO, PARM_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
</tbody>
</table>
### Business Flag

This table describes the Business Flag maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type</th>
<th>Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_BUS_FLG</td>
<td>RANGE (ILM_DT,BUS_FLG_ID)</td>
<td>F1T681P0</td>
<td>BUS_FLG_ID</td>
<td>Global</td>
<td></td>
<td>RANGE(BUS_FLG_ID)</td>
<td>F1_BUS_FLG_CRE_DTTM</td>
</tr>
<tr>
<td>(Parent)</td>
<td></td>
<td>F1T681S1</td>
<td>BUS_OBJ_CD, BO_STATUS_CD, BUS_FLG_ID</td>
<td>Global</td>
<td></td>
<td>RANGE(BUS_FLG_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CM_ILM_F1T681S2</td>
<td>ILM_DT, ILM_ARCH_SW, BUS_FLG_ID</td>
<td>Local</td>
<td></td>
<td>RANGE(BUS_FLG_ID)</td>
<td></td>
</tr>
<tr>
<td>F1_BUS_FLG_CHAR</td>
<td>Reference Partitioning</td>
<td>F1T684P0</td>
<td>BUS_FLG_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td></td>
<td>RANGE(BUS_FLG_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T684S0</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td>RANGE(BUS_FLG_ID)</td>
<td></td>
</tr>
<tr>
<td>F1_BUS_FLG_LOG</td>
<td>Reference Partitioning</td>
<td>F1T685P0</td>
<td>BUS_FLG_ID, SEQNO</td>
<td>Global</td>
<td></td>
<td>RANGE(BUS_FLG_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T685S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td>Global</td>
<td></td>
<td>RANGE(BUS_FLG_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T685S2</td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td>Global</td>
<td></td>
<td>RANGE(BUS_FLG_ID)</td>
<td></td>
</tr>
<tr>
<td>F1_BUS_FLG_LOG_PARM</td>
<td>Reference Partitioning</td>
<td>F1T686P0</td>
<td>BUS_FLG_ID, SEQNO, PARM_SEQ</td>
<td>Global</td>
<td></td>
<td>RANGE(BUS_FLG_ID)</td>
<td></td>
</tr>
<tr>
<td>F1_BUS_FLG_REL</td>
<td>Reference Partitioning</td>
<td>F1T682P0</td>
<td>BUS_FLG_ID, BUS_FLG_REL_TYPE_FLG, SEQ_NUM</td>
<td>Global</td>
<td></td>
<td>RANGE(BUS_FLG_ID)</td>
<td></td>
</tr>
<tr>
<td>F1_BUS_FLG_REL_OBJ</td>
<td>Reference Partitioning</td>
<td>F1T683P0</td>
<td>BUS_FLG_ID, BUS_FLG_REL_OBJ_TYPE_FLG, SEQ_NUM</td>
<td>Global</td>
<td></td>
<td>RANGE(BUS_FLG_ID)</td>
<td></td>
</tr>
</tbody>
</table>
Remote Message  
This table describes the Remote Message maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Type</th>
<th>Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_REMOTE_MSG</td>
<td>RANGE (ILM_DT,F1_REMOTE_MSG_ID)</td>
<td>F1T735P0</td>
<td>F1_REMOTE_MSG_ID</td>
<td>Global Partitioned</td>
<td>RANGE(F1_REMOTE_MSG_ID)</td>
<td>F1_REMOTE_MSG.CRE_DTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T735S1</td>
<td>CRE_DTTM</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T735S2</td>
<td>F1_MDT_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T735S3</td>
<td>MAINT_OBJ_CD</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T735S4</td>
<td>PK_VALUE1</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T735S5</td>
<td>F1_DEVICE_MSG_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T735S6</td>
<td>F1_MDT_ID, F1_MSG_CLASS_FLG, F1_DELIVERY_STATUS_FLG</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM</td>
<td>ILM_DT, ILM_ARCH_SW, F1_REMOTE_MSG_ID</td>
<td>F1T735S7</td>
<td>ILM_DT</td>
<td>Local Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_REMOTE_MSG</td>
<td>Reference Partitioning</td>
<td>F1T736P0</td>
<td>F1_REMOTE_MSG_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T736S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_REMOTE_MSG</td>
<td>Reference Partitioning</td>
<td>F1T737P0</td>
<td>F1_REMOTE_MSG_ID, SEQNO</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T737S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T737S2</td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_REMOTE_MSG</td>
<td>Reference Partitioning</td>
<td>F1T738P0</td>
<td>F1_REMOTE_MSG_ID, SEQNO, PARM_SEQ</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Statistics Snapshot

This table describes the Statistics Snapshot maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_STATS_SNAPSHOT( Parent)</td>
<td>RANGE (ILM_DT, SNAPSHOT_ID)</td>
<td>F1C706P0</td>
<td>SNAPSHOT_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>F1C706S1</td>
</tr>
<tr>
<td>BM_STATS_SNAPSHOT</td>
<td>RANGE (SNAPSHOT_ID)</td>
<td>F1C706S1</td>
<td>BUS_OBJ_CD, BO_STATUS_CD, SNAPSHOT_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_F1C706S2</td>
<td>ILM_DT, ILM_ARCH_SW, SNAPSHOT_ID</td>
<td>Local</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_STATS_SNAPSHOT_CHAR</td>
<td>Reference Partitioning</td>
<td>F1C707P0</td>
<td>SNAPSHOT_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td>F1C707S1</td>
</tr>
<tr>
<td>BM_STATS_SNAPSHOT_REL_OBJ</td>
<td>Reference Partitioning</td>
<td>F1C708P0</td>
<td>SNAPSHOT_ID, SEQNO</td>
<td>Global</td>
<td>Partitioned</td>
<td>F1C708S1</td>
</tr>
<tr>
<td>F1C708S2</td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1C709P0</td>
<td>SNAPSHOT_ID, SEQNO, PARM_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1C710P0</td>
<td>SNAPSHOT_ID, STATS_SNAPSHOT_REL_OBJ, PARM_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Customer Relationship Request
This table describes the Customer Relationship Request maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type</th>
<th>Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1_CUST_REL_REQ</td>
<td>RANGE/ILM_DT (CUST_REL_REQ_ID)</td>
<td>CIT017P0</td>
<td>CUST_REL_REQ_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>C1_CUST_REL_REQ.CRE_DTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIT017S1</td>
<td>CUST_REL_REQ_ID, BUS_OBJ_CD, BO_STATUS_CD</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_CIT017S2</td>
<td>ILM_DT, ILM_ARCH_SW, CUST_REL_REQ_ID</td>
<td>CM_ILM_CIT017S2</td>
<td>ILM_DT, ILM_ARCH_SW, CUST_REL_REQ_ID</td>
<td>Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1_CUST_REL_REQ_CHAR</td>
<td>Reference Partitioning</td>
<td>CIT014P0</td>
<td>CUST_REL_REQ_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIT014S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1_CUST_REL_REQ_LOG</td>
<td>Reference Partitioning</td>
<td>CIT015P0</td>
<td>CUST_REL_REQ_ID, SEQNO</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIT015S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIT015S2</td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1_CUST_REL_REQ_LOG_PARM</td>
<td>Reference Partitioning</td>
<td>CIT016P0</td>
<td>CUST_REL_REQ_ID, SEQNO, PARM_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIT018P0</td>
<td>CUST_REL_REQ_ID, CRR_REL_OBJ_TYPE_FLG, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
</tbody>
</table>
### Notification Communication Preference

This table describes the Notification Communication Preference maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1_NTF_PREF</td>
<td>RANGE(ILM_DT, NTF_PREF_ID)</td>
<td>CIT002P0</td>
<td>NTF_PREF_ID</td>
<td>Global Partitioned</td>
<td>C1_NTF_PREF_STATUS_UPD_DTTM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIT002S1</td>
<td>ACCT_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIT002S2</td>
<td>NTF_PREF_ID,</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BO_STATUS_CD,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>STATUS_UPD_DTTM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIT002S3</td>
<td>C1_CONTACT_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIT002S4</td>
<td>F1_SVC_TASK_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CM_ILM_CIT002S3</td>
<td>ILM_DT, ILM_ARCH_SW, NTF_PREF_ID</td>
<td>Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1_NTF_PREF_CHAR</td>
<td>Reference Partitioning</td>
<td>CIT003P0</td>
<td>NTF_PREF_ID,</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CHAR_TYPE_CD,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SEQ_NUM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIT003S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1_NTF_PREF_ID</td>
<td>Reference Partitioning</td>
<td>CIT006P0</td>
<td>NTF_PREF_ID,</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NTF_PREF_ID_TY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PE_FLG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1_NTF_PREF_LOG</td>
<td>Reference Partitioning</td>
<td>CIT004P0</td>
<td>NTF_PREF_ID,</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SEQNO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIT004S1</td>
<td>CHAR_TYPE_CD,</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CHAR_VAL_FK1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIT004S2</td>
<td>CHAR_TYPE_CD,</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CHAR_VAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1_NTF_PREF_LOG_PARM</td>
<td>Reference Partitioning</td>
<td>CIT005P0</td>
<td>NTF_PREF_ID,</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SEQNO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PARM_SEQ</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Customer Contact
This table describes the Customer Contact maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_CC</td>
<td>RANGE(CC_DT_TM,CC_ID)</td>
<td>XT057P0</td>
<td>CC_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>RANGE(CC_ID)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT057S1</td>
<td>PER_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT057S2</td>
<td>BATCH_CD, BATCH_NBR, CC_ID, PRINT_LETTER_SW</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CC_TYPE_CD</td>
<td>CC_TYPE_CD, CC_CL_CD, CC_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT057S4</td>
<td>ACCT_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT057S5</td>
<td>PREM_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_CC_CHAR</td>
<td>Reference Partitioning</td>
<td>XT280P0</td>
<td>CC_ID, CHAR_TYPE_CD</td>
<td>Global</td>
<td>Partitioned</td>
<td>RANGE(CC_ID)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C1T007S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_CC_LOG</td>
<td>Reference Partitioning</td>
<td>XT281P0</td>
<td>CC_LOG_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>RANGE(CC_LOG_ID)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT281S1</td>
<td>CC_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Collection Process
This table describes the Collection Process maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_COLL_PROC</td>
<td>RANGE(CRE_D TTM,COLL_PROC_ID)</td>
<td>XT073P0</td>
<td>COLL_PROC_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>RANGE(COLL_PROC_ID)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT073S1</td>
<td>ACCT_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Cut Process

This table describes the Cut Process maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_COLL_EVT</td>
<td>Reference Partitioning</td>
<td>XT069P0</td>
<td>COLL_PROC_ID, EVT_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_COLL_EVT_CC</td>
<td>Reference Partitioning</td>
<td>XT070P0</td>
<td>COLL_PROC_ID, EVT_SEQ, CC_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_COLL(proc SA)</td>
<td>Reference Partitioning</td>
<td>XT074P0</td>
<td>COLL_PROC_ID, SA_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_CUT_PROC</td>
<td>RANGE(CRE_DT TTM, CUT_PROC_ID)</td>
<td>XT324P0</td>
<td>CUT_PROC_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_CUT_EVT</td>
<td>Reference Partitioning</td>
<td>XT322P0</td>
<td>CUT_PROC_ID, EVT_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_CUT_EVT_DEP</td>
<td>Reference Partitioning</td>
<td>XT323P0</td>
<td>CUT_PROC_ID, EVT_SEQ, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
</tbody>
</table>
# Financial Transaction

This table describes the Financial Transaction maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_FT</td>
<td>RANGE(CRE_D, TTM, FT_ID)</td>
<td>XT112P0</td>
<td>FT_ID</td>
<td>Global Partitioned</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT112S1</td>
<td>SA_ID, REDUNDANT_SW, BILL_ID</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT112S2</td>
<td>BILL_ID</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT112S3</td>
<td>PARENT_ID</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT112S4</td>
<td>SIBLING_ID</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT112S5</td>
<td>MATCH_EVT_ID, FT_ID</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT112S6</td>
<td>GL_DISTRIB_, STATUS, SCHED_DISTRIB_, DT, FT_ID</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT112S7</td>
<td>BAL_CTL_GRP_ID, FT_ID</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td>CI_FT_GL</td>
<td>Reference Partitioning</td>
<td>XT113P0</td>
<td>FT_ID, GL_SEQ_NBR</td>
<td>Global Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT113S1</td>
<td>GL_ACCT, FT_ID</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td>CI_FT_PROC</td>
<td>Reference Partitioning</td>
<td>XT097P0</td>
<td>FT_ID, SEQ_NUM</td>
<td>Global Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT097S1</td>
<td>BATCH_CD, BATCH_NBR</td>
<td>Global</td>
<td></td>
</tr>
</tbody>
</table>
## Overdue Process
This table describes the Overdue Process maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_OD_PROC</td>
<td>RANGE(CRE_D TTM,OD_PROC_ID)</td>
<td>XT315P0</td>
<td>OD_PROC_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>N/A</td>
</tr>
<tr>
<td>CI_OD_EVT</td>
<td>Reference Partitioning</td>
<td>XT318P0</td>
<td>OD_PROC_ID, EVT_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_OD_EVT_DEP</td>
<td>Reference Partitioning</td>
<td>XT319P0</td>
<td>OD_PROC_ID, EVT_SEQ, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_OD_PROC_LOG</td>
<td>Reference Partitioning</td>
<td>XT320P0</td>
<td>OD_PROC_ID, LOG_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT320S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_OD_PROC_LOGPARM</td>
<td>Reference Partitioning</td>
<td>XT321P0</td>
<td>OD_PROC_ID, LOG_SEQ, PARM_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_OD_PROC_OBJ</td>
<td>Reference Partitioning</td>
<td>XT317P0</td>
<td>OD_PROC_ID, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
</tbody>
</table>

## Severance Event
This table describes the Severance Event maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_SEV_PROC</td>
<td>RANGE(CRE_D TTM,SEV_PROC_ID)</td>
<td>XT118P0</td>
<td>SEV_PROC_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT118S1</td>
<td>SA_ID, SEV_PROC_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT118S2</td>
<td>COLL_PROC_ID, EVT_SEQ</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_SEV_EVT</td>
<td>Reference Partitioning</td>
<td>XT214P0</td>
<td>SEV_PROC_ID, EVT_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
</tbody>
</table>
## Naming Convention

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Type</th>
<th>Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_SEV_EVT_DEP</td>
<td>Reference Partitioning</td>
<td>XT216P0</td>
<td>SEV_PROC_ID, EVT_SEQ, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_SEV_EVT_FA</td>
<td>Reference Partitioning</td>
<td>XT217P0</td>
<td>SEV_PROC_ID, EVT_SEQ, FA_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT217S1</td>
<td>FA_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_SEV_EVT_CC</td>
<td>Reference Partitioning</td>
<td>XT215P0</td>
<td>SEV_PROC_ID, EVT_SEQ, CC_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT215S1</td>
<td>CC_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### WO Process

This table describes the WO Process maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Type</th>
<th>Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_WO_PROC</td>
<td>RANGE(CRE_DTTM, WO_PROC_ID)</td>
<td>XT061P0</td>
<td>WO_PROC_ID</td>
<td>Global</td>
<td>(WO_PROC_ID)</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT061S1</td>
<td>ACCT_ID, WO_STATUS_FLG</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_WO_EVT</td>
<td>Reference Partitioning</td>
<td>XT059P0</td>
<td>WO_PROC_ID, EVT_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CI_WO_EVT_CC</td>
<td>Reference Partitioning</td>
<td>XT060P0</td>
<td>WO_PROC_ID, EVT_SEQ, CC_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT060S1</td>
<td>CC_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_WO_PROC_SA</td>
<td>Reference Partitioning</td>
<td>XT062P0</td>
<td>WO_PROC_ID, SA_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT062S1</td>
<td>SA_ID, WO_SA_STAT_FLG</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## General Audit

This table describes the General Audit maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_GNRL_AUDIT</td>
<td>RANGE(ILM_DT, AUDIT_ID)</td>
<td>F1T901P0</td>
<td>AUDIT_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>F1_GNRL_AUDIT T.CRE_DTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T901S1</td>
<td>USER_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T901S1</td>
<td>AUDIT_ID, USER_ID, CRE_DTTM</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_F1T901S3</td>
<td>ILM_DT, ILM_ARCH_SW, AUDIT_ID</td>
<td></td>
<td></td>
<td>Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_GNRL_AUDIT T_CHAR</td>
<td>Reference Partitioning</td>
<td>F1C504P0</td>
<td>AUDIT_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C504S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_GNRL_AUDIT T_VAL</td>
<td>Reference Partitioning</td>
<td>F1T902P0</td>
<td>AUDIT_ID, FLD_NAME</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T902S1</td>
<td>AUDIT_ID, FLD_NAME, FLD_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Oracle Utilities Customer To Meter provides support for Information Lifecycle Management (ILM) and Data Archiving.

ILM is a process 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 passes the data is accessed less frequently or not at all. The costs of storing data that are accessed infrequently can be reduced by moving the data to lower cost mass storage media. Typically this involves a trade-off between cost and increased access times. Based on business needs, data may eventually be archived and purged from the database and kept offline ready to be restored if required.

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 application and database configuration and requires Oracle Partitioning.

An underlying design principle of the Oracle Utilities Application Framework ILM implementation is the concept that the age of the data may not be the only criterion used to determine when a record is able to be archived. There may be business rules that dictate that some records are still current and must not be archived yet.

ILM enabled objects have a combination of an ILM date and an ILM Archive Switch. The ILM date is used in conjunction with partitioning to group data by age. The ILM Archive Switch is set by a background process when the record meets the business rules specific to that Maintenance Object if the record is eligible to be archived. The ILM Archive Switch gives Database Administrators an easy method to check when all records in a partition meet the business criteria that make the partition eligible to be archived. If the ILM Archive Switch is set for all records, then the DBA can take the steps required to archive the partition.

Moving data between storage tiers takes advantage of the partitioning by ILM Date but does not require that the ILM Archive Switch is set. Oracle recommends using the Oracle Database ILM Assistant to assist with this process.

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 date column is defaulted to an appropriate date (typically the system date) when a new record is inserted, the MO is partitioned on the ILM_DT, so it should only be updated in exceptional circumstances as this would cause the record to be deleted from its current partition and inserted into a different partition, which is a relatively expensive operation.
  - **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 indicating that the record is eligible to be archived.
- **Database Referential Integrity Constraints** - These are required for reference partitioning of Child tables of ILM enabled MOs.
- **Partitioning** - Partitioning is mandatory for ILM implementation. It is used to separate the data by ILM date so that data of a similar age is kept together.
- **One Tablespace per Partition** - The ILM implementation requires that each MO partition resides in a dedicated tablespace so that they can be easily managed.
- **Naming Convention** - This section covers the recommended naming convention to be used for partitions/subpartitions and tablespaces.
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**

Note: In order to successfully implement ILM as described here, the following DB Version and Patch are pre-requisites: database version 12.1.0.2.0 Enterprise Edition and Patch 15996848.

The steps needed to enable ILM functionality differ depending on whether ILM is enabled as part of the initial implementation of the product or enabled ILM on an existing implementation where data already exists in the respective tables.

- **Initial Install** – For an initial installation, the section Module Specific ILM Implementation Details outlines the additional steps to be performed on base delivered ILM Enabled Tables to conform to ILM requirements. In addition, Appendix A: Sample SQL for Enabling ILM in MDM (Initial Installation) provides sample reference DDLs using two maintenance objects as examples.

- **Transform NON-ILM implementation to ILM Enabled Implementation:** The following steps provide a high level overview of steps that must be performed to implement ILM on enabled MOs for an existing implementation. Please refer to the Appendix B: Sample SQL for Enabling ILM in MDM (Existing Installation) section for detailed information using To Do Entry as an example. Also refer to Appendix C: Sample SQL for Enabling ILM with Sub Retention in MDM (Existing Installation) or detailed information using D1_INIT_MSRMT_DATA as an example.

1. Rename the existing tables (Parent table followed by child table), 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’.

**Note:** Certain ILM enabled MOs, specifically IMD, Device Event, and Activity, support more than one retention period also known as sub-retention periods. For these MOs the table will be sub-partitioned based on the retention period. Furthermore, a more detailed approach will be...
required to set both the ILM date (ILM_DT) and the retention period (<field name>). If your implementation does not wish to leverage the ability to define multiple retention periods for these MOs, this note can be ignored and the general guidelines for ILM enablement can be followed. If your implementation wishes to leverage the multiple retention period capability then please refer to the section Module Specific ILM Implementation Details For Sub Retention below.

5. Enable logging option.
6. Create Primary Key index.
7. Create Primary Key Constraint 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 a 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 B: Sample SQL for Enabling ILM in MDM (Existing Installation). If sub-retention is supported, sample DDL for child tables can be found in Appendix C: Sample SQL for Enabling ILM with Sub Retention in MDM (Existing Installation).

   Please note that child table should be partitioned using reference partitioning of the parent table’s partitioning key.

10. Drop the original, renamed tables after verifying the newly created partitioned tables.
11. If sub-retention is not supported, create the ILM specific indexes from section Module Specific ILM Implementation Details.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Index Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_TD_ENTRY</td>
<td>CM_ILM_XT039S8</td>
</tr>
<tr>
<td>D1_ACTIVITY</td>
<td>CM_ILM_D1T319S1</td>
</tr>
<tr>
<td>D1_COMM_IN</td>
<td>CM_ILM_D1T386S1</td>
</tr>
<tr>
<td>D1_COMM_OUT</td>
<td>CM_ILM_D1T380S1</td>
</tr>
<tr>
<td>D1_COMPL_EVT</td>
<td>CM_ILM_D1T340S1</td>
</tr>
<tr>
<td>D1_DVC_EVT</td>
<td>CM_ILM_D1T400S4</td>
</tr>
<tr>
<td>D1_INIT_MSRMT_DATA</td>
<td>CM_ILM_D1T304S4</td>
</tr>
<tr>
<td>D1_USAGE</td>
<td>CM_ILM_D1T281S2</td>
</tr>
<tr>
<td>D1_USAGE_EXCP</td>
<td>CM_ILM_D1T443S1</td>
</tr>
<tr>
<td>D1_VEE_EXCP</td>
<td>CM_ILM_D1T308S2</td>
</tr>
<tr>
<td>D1_SNAPSHOT_DL_CTRL</td>
<td>CM_ILM_D1T433S1</td>
</tr>
<tr>
<td>D1_SP_SNAP_DL</td>
<td>CM_ILM_D1T434S1</td>
</tr>
<tr>
<td>D1_SP_UNR_USG_SNAP_DL</td>
<td>CM_ILM_D1T438S1</td>
</tr>
</tbody>
</table>
12. If sub-retention is supported, create the following ILM specific indexes from the Module Specific ILM Implementation Details section:

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Index Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_SP_USG_SNAP_DL</td>
<td>CM_ILM_D1T436S1</td>
</tr>
<tr>
<td>D1_SP_VEE_EXCP_SNAP_DL</td>
<td>CM_ILM_D1T440S1</td>
</tr>
<tr>
<td>F1_BUS_FLG</td>
<td>CM_ILM_F1T681S2</td>
</tr>
<tr>
<td>F1_OBJ_REV</td>
<td>CM_ILM_FT035S6</td>
</tr>
<tr>
<td>F1_OUTMSG</td>
<td>CM_ILM_FT010S2</td>
</tr>
<tr>
<td>F1_REMOTE_MSG</td>
<td>CM_ILM_F1T735S7</td>
</tr>
<tr>
<td>F1_STATS_SNPSHT</td>
<td>CM_ILM_F1C706S2</td>
</tr>
<tr>
<td>F1_SVC_TASK</td>
<td>CM_ILM_F1C474S3</td>
</tr>
<tr>
<td>F1_SYNC_REQ</td>
<td>CM_ILM_F1T014S4</td>
</tr>
<tr>
<td>F1_SYNC_REQ_IN</td>
<td>CM_ILM_F1T191S3</td>
</tr>
<tr>
<td>CI_TD_ENTRY</td>
<td>CM_ILM_XT039S8</td>
</tr>
<tr>
<td>D1_COMM_IN</td>
<td>CM_ILM_D1T386S1</td>
</tr>
<tr>
<td>D1_COMM_OUT</td>
<td>CM_ILM_D1T380S1</td>
</tr>
<tr>
<td>D1_COMPL_EVT</td>
<td>CM_ILM_D1T340S1</td>
</tr>
<tr>
<td>D1_USAGE</td>
<td>CM_ILM_D1T281S2</td>
</tr>
<tr>
<td>D1_USAGE_EXCP</td>
<td>CM_ILM_D1T443S1</td>
</tr>
<tr>
<td>D1_VEE_EXCP</td>
<td>CM_ILM_D1T308S2</td>
</tr>
<tr>
<td>D1_SNAPSHOT_DL_CTRL</td>
<td>CM_ILM_D1T433S1</td>
</tr>
<tr>
<td>D1_SP_SNAP_DL</td>
<td>CM_ILM_D1T434S1</td>
</tr>
<tr>
<td>D1_SP_UNR_USG_SNAP_DL</td>
<td>CM_ILM_D1T438S1</td>
</tr>
<tr>
<td>D1_SP_USG_SNAP_DL</td>
<td>CM_ILM_D1T436S1</td>
</tr>
<tr>
<td>D1_SP_VEE_EXCP_SNAP_DL</td>
<td>CM_ILM_D1T440S1</td>
</tr>
<tr>
<td>F1_BUS_FLG</td>
<td>CM_ILM_F1T681S2</td>
</tr>
<tr>
<td>F1_OBJ_REV</td>
<td>CM_ILM_FT035S6</td>
</tr>
<tr>
<td>F1_OUTMSG</td>
<td>CM_ILM_FT010S2</td>
</tr>
<tr>
<td>F1_REMOTE_MSG</td>
<td>CM_ILM_F1T735S7</td>
</tr>
<tr>
<td>F1_STATS_SNPSHT</td>
<td>CM_ILM_F1C706S2</td>
</tr>
<tr>
<td>F1_SVC_TASK</td>
<td>CM_ILM_F1C474S3</td>
</tr>
<tr>
<td>F1_SYNC_REQ</td>
<td>CM_ILM_F1T014S4</td>
</tr>
</tbody>
</table>
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.

All indexes are listed with a recommendation whether the index should be global or local and whether the index should be partitioned. In addition to the base delivered indexes, each 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. The recommended column that should be used to populate the ILM_DT is also shown.

This section details the following maintenance objects:

- To Do Entry
- Sync Request (Outbound)
- Inbound Sync Request
- Outbound Message
- Service Task
- Object Revision
- Business Flag
- Remote Message
- Statistics Snapshot
- Activity
- Communication In
- Communication Out
- Device Event
- Completion Event
- Initial Measurement Data
- Usage Transaction
- Usage Transaction Exception
- VEE Exception

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Index Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_SYNC_REQ_IN</td>
<td>CM_ILM_F1T191S3</td>
</tr>
<tr>
<td>D1_ACTIVITY</td>
<td>CM_ILM_D1T319S1</td>
</tr>
<tr>
<td>D1_DVC_EVT</td>
<td>CM_ILM_D1T400S4</td>
</tr>
<tr>
<td>D1_INIT_MSRMT_DATA</td>
<td>CM_ILM_D1T304S4</td>
</tr>
</tbody>
</table>

and the ILM subretention specific indexes from the Module Specific ILM Implementation Details For Sub Retention section:
To Do Entry
This table describes the To Do Entry maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_TD_ENTRY (Parent)</td>
<td>RANGE (ILM_DT, TD_ENTRY_ID)</td>
<td>XT039P0</td>
<td>TD_ENTRY_ID</td>
<td>Global</td>
<td>RANGE (TD_ENTRY_ID)</td>
<td>CI_TD_ENTRY, CRE_DTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT039S2</td>
<td>ASSIGNED_TO, TD_ENTRY_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT039S3</td>
<td>ENTRY_STATUS_FLG, ASSIGNED_TO</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT039S4</td>
<td>ROLE_ID, TD_TYPE_CD, ENTRY_STATUS_FLG, TD_PRIORITY_FLG</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT039S5</td>
<td>BATCH_CD, BATCH_NBR, ENTRY_STATUS_FLG</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT039S6</td>
<td>TD_ENTRY_ID, ASSIGNED_TO, ENTRY_STATUS_FLG</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT039S7</td>
<td>COMPLETE_USER_ID, COMPLETE_DTTM, TD_ENTRY_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_ XT039S8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_TD_ENTRY_</td>
<td>Reference Partitioning</td>
<td>XT701P0</td>
<td>TD_ENTRY_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CHA</td>
<td></td>
<td>XT701S1</td>
<td>SRCH_CHAR_VAL, CHAR_TYPE_CD, TD_ENTRY_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT701S2</td>
<td>CHAR_VAL_FK1</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_TD_DRLKEY</td>
<td>Reference Partitioning</td>
<td>XT037P0</td>
<td>TD_ENTRY_ID, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT037S1</td>
<td>KEY_VALUE, TD_ENTRY_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI_TD_LOG</td>
<td>Reference Partitioning</td>
<td>XT721P0</td>
<td>TD_ENTRY_ID, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>Table Name</td>
<td>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</td>
<td>Index Name</td>
<td>Index Columns</td>
<td>Index Type Global or Local</td>
<td>Index Partitioning Sub-Partitioning Key</td>
<td>ILM_DT Initial Load</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------</td>
<td>------------</td>
<td>-----------------------------------</td>
<td>---------------------------</td>
<td>----------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>XT721S1</td>
<td>LOG_DTTM,USER_ID, LOG_TYPE_FLG, TD_ENTRY_ID</td>
<td></td>
<td></td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CL_TD_MSG_parms</td>
<td>Reference Partitioning</td>
<td>XT040P0</td>
<td>TD_ENTRY_ID, SEQ_NUM</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CL_TD_SRTKEY</td>
<td>Reference Partitioning</td>
<td>XT041P0</td>
<td>TD_ENTRY_ID, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XT041S1</td>
<td>KEY_VALUE, TD_ENTRY_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sync Request (Outbound)

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

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_SYNC_REQ</td>
<td>RANGE (ILM_DT, F1_SYNC_REQ_ID)</td>
<td>FIT014P0</td>
<td>F1_SYNC_REQ_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FIT014S1</td>
<td>BO_STATUS_CD, BUS_OBJ_CD, F1_SYNC_REQ_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FIT014S2</td>
<td>BO_STATUS_REASON_CD</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FIT014S3</td>
<td>MAINT_OBJ_CD, PK_VALUE1, PK_VALUE2, F1_SYNC_REQ_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table Name</td>
<td>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</td>
<td>Index Name</td>
<td>Index Columns</td>
<td>Index Type</td>
<td>Local/Global or Local</td>
<td>Sub-Partitioning Key</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------------------------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>------------</td>
<td>-----------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>CM_ILM_FIT014 S4</td>
<td></td>
<td></td>
<td>ILM_DT, ILM_ARC_SW, F1_SYNC_REQ_ID</td>
<td></td>
<td>Local</td>
<td></td>
</tr>
<tr>
<td>F1_SYNC_REQ_CHAR</td>
<td>Reference Partitioning</td>
<td>FIT017P0</td>
<td>F1_SYNC_REQ_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td></td>
<td>Global</td>
<td>Partitioned</td>
</tr>
<tr>
<td>F1_SYNC_REQ_EXTRACT</td>
<td>Reference Partitioning</td>
<td>FIT019P0</td>
<td>F1_SYNC_REQ_ID, SEQ_NUM</td>
<td></td>
<td>Global</td>
<td>Partitioned</td>
</tr>
<tr>
<td>F1_SYNC_REQ_LOG</td>
<td>Reference Partitioning</td>
<td>FIT015P0</td>
<td>F1_SYNC_REQ_ID, SEQNO</td>
<td></td>
<td>Global</td>
<td>Partitioned</td>
</tr>
<tr>
<td>F1_SYNC_REQ_LOG_PARM (Child Table of F1_SYNC_REQ_LOG_PARM)</td>
<td>Reference Partitioning</td>
<td>FIT016P0</td>
<td>F1_SYNC_REQ_ID, SEQNO, PARM_SEQ</td>
<td></td>
<td>Global</td>
<td>Partitioned</td>
</tr>
</tbody>
</table>

**Note:** It is recommended that data retention policies and rules for this object match the policies and rules implemented for the Inbound Sync Request on the target system to avoid data inconsistencies when auditing.
# Inbound Sync Request
This table describes the Inbound Sync Request maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_SYNC_REQ_IN_IN</td>
<td>RANGE(ILM_DT, F1_SYNC_REQ_IN_ID)</td>
<td>F1T191P0</td>
<td>F1_SYNC_REQ_IN_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>F1_SYNC_REQ_IN_CRE_DTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T191S1</td>
<td>BO_STATUS_CD, BUS_OBJ_CD, F1_SYNC_REQ_IN_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T191S2</td>
<td>MAINT_OBJ_CD, EXT_PK_VALUE1, NT_XID_CD, PK_VALUE1</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CM_ILM_F1T191S3</td>
<td>ILM_DT, ILM_ARCH_Sw, F1_SYNC_REQ_IN_ID</td>
<td>Local</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reference Partitioning</td>
<td>F1T193P0</td>
<td>F1_SYNC_REQ_IN_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>F1_SYNC_REQ_IN_CHAR</td>
<td>Reference Partitioning</td>
<td>F1T193S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reference Partitioning</td>
<td>F1T197P0</td>
<td>F1_SYNC_REQ_IN_ID, SEQNO</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reference Partitioning</td>
<td>F1T198P0</td>
<td>F1_SYNC_REQ_IN_ID, SEQNO, PARM_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reference Partitioning</td>
<td>F1T194P0</td>
<td>F1_SYNC_REQ_IN_ID, SEQNO</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reference Partitioning</td>
<td>F1T194S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reference Partitioning</td>
<td>F1T194S2</td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Outbound Message

This table describes the Outbound Message maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>F1_OUTMSG</strong> (Parent)</td>
<td>RANGE (ILM_DT, OUTMSG_ID)</td>
<td>FT010P0</td>
<td>OUTMSG_ID</td>
<td>RANGE (OUMSG_ID)</td>
<td>F1_OUTMSG, CRE_DTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT010S1</td>
<td>OUTMSG_STAT, US_FLG, OUTMSG_TYPE_CD</td>
<td>Global</td>
<td></td>
</tr>
<tr>
<td>CM_ILM_FT010S</td>
<td>ILM_DT, ILM_ARC_SW, OUTMSG_ID</td>
<td>Local</td>
<td>Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_OUTMSG_ERRPARM</td>
<td>Reference Partitioning</td>
<td>FT011P0</td>
<td>OUTMSG_ID, PARM_SEQ</td>
<td>Global</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** It is recommended that data retention policies and rules for this object match the policies and rules implemented for the Outbound Sync Request on the source system to avoid data inconsistencies when auditing.
## Service Task

This table describes the Service Task maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_SVC_TASK (Parent)</td>
<td>RANGE (ILM_DT, F1_SVC_TASK_ID)</td>
<td>F1C474P0</td>
<td>F1_SVC_TASK_ID</td>
<td>Global Partitioned</td>
<td>RANGE (F1_SVC_TASK_CRE_DTTM)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C474S1</td>
<td>F1_SVC_TASK_ID</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C474S2</td>
<td>BUS_OBJ_CD</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CM_ILM_F1C474S2</td>
<td></td>
<td>ILM_DT, ILM_ARC_SW, F1_SVC_TASK_ID</td>
<td>Local Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_SVC_TASK_CHAR</td>
<td>Reference Partitioning</td>
<td>F1C476P0</td>
<td>F1_SVC_TASK_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C476S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_SVC_TASK_LOG</td>
<td>Reference Partitioning</td>
<td>F1C477P0</td>
<td>F1_SVC_TASK_ID, SEQNO</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C477S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>F1C477S2</td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_SVC_TASK_LOG_PARM (Child Table of F1_SVC_TASK_LOG)</td>
<td>Reference Partitioning</td>
<td>F1C478P0</td>
<td>F1_SVC_TASK_ID, SEQNO, PARM_SEQ</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C479P0</td>
<td>F1_SVC_TASK_ID, MAINT_OBJ_CD, SEQ_NUM</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C479S1</td>
<td>MAINT_OBJ_CD, PK_VALUE1, PK_VALUE2, PK_VALUE3</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Object Revision
This table describes the Object Revision maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_OBJ_REV</td>
<td>RANGE (ILM_DT, REV_ID)</td>
<td>FT035P0</td>
<td>REV_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT035S1</td>
<td>BO_STATUS_CD, BUS_OBJ_CD, REV_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT035S2</td>
<td>MAINT_OBJ_CD, PK_VALUE1</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT035S3</td>
<td>EXT_REFERENCE_ID, MAINT_OBJ_CD</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT035S4</td>
<td>USER_ID, MAINT_OBJ_CD</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FT035S5</td>
<td>PK_VALUE1</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CM_ILM_FT035S6</td>
<td></td>
<td>ILM_DT, ILM_ARC_SW, REV_ID</td>
<td>Local</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>FT037P0</td>
<td>Reference Partitioning</td>
<td>FT037P0</td>
<td>REV_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>FT037S1</td>
<td>Reference Partitioning</td>
<td></td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FT039P0</td>
<td>Reference Partitioning</td>
<td></td>
<td>REV_ID, SEQNO</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>FT040P0</td>
<td>Reference Partitioning</td>
<td></td>
<td>REV_ID, SEQNO, PARM_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
</tbody>
</table>

**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.
**Business Flag**
This table describes the Business Flag maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_BUS_FLG (Parent)</td>
<td>RANGE (ILM_DT,BUS_FLG_ID)</td>
<td>F1T681P0</td>
<td>BUS_FLG_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>RANGE(BUS_FLG_ID)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T681S1</td>
<td>BUS_OBJ_CD, BO_STATUS_CD, BUS_FLG_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CM_ILM_F1T681S2</td>
<td>ILM_DT, ILM_ARCH_SW, BUS_FLG_ID</td>
<td>Local</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>F1_BUS_FLG_CHAR</td>
<td>Reference Partitioning</td>
<td>F1T684P0</td>
<td>BUS_FLG_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T684S0</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_BUS_FLG_LOG</td>
<td>Reference Partitioning</td>
<td>F1T685P0</td>
<td>BUS_FLG_ID, SEQNO</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T685S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T685S2</td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_BUS_FLG_LOG_PARM</td>
<td>Reference Partitioning</td>
<td>F1T686P0</td>
<td>BUS_FLG_ID, SEQNO, PARM_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>F1_BUS_FLG_REL</td>
<td>Reference Partitioning</td>
<td>F1T682P0</td>
<td>BUS_FLG_ID, BUS_FLG_REL_TYPE_FLG, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>F1_BUS_FLG_REL_OBJ</td>
<td>Reference Partitioning</td>
<td>F1T683P0</td>
<td>BUS_FLG_ID, BUS_FLG_REL_OBJ_TYPE_FLG, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
</tbody>
</table>
Remote Message
This table describes the Remote Message maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_REMOTE_MSG (Parent)</td>
<td>RANGE (ILM_DT,F1_REMOTE_MSG_ID)</td>
<td>F1T735P0</td>
<td>F1_REMOTE_MSG_ID</td>
<td>Global</td>
<td>RANGE(F1_REMOTE_MSG_ID)</td>
<td>F1_REMOTE_MSG.CRE_DTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T735S1</td>
<td>CRE_DTTM</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T735S2</td>
<td>F1_MDT_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T735S3</td>
<td>MAINT_OBJ_CD</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T735S4</td>
<td>PK_VALUE1</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T735S5</td>
<td>F1_DEVICE_MSG_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T735S6</td>
<td>F1_MDT_ID,</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>F1_MSG_CLASS_FLG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>F1_DELIVERY_STATE_FLG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_F1T735S7</td>
<td>ILM_DT, ILM_ARCH_SW, F1_REMOTE_MSG_ID</td>
<td>F1T736P0</td>
<td>F1_REMOTE_MSG_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>RANGE(F1_REMOTE_MSG_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T736S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_REMOTE_MSG_CHAR</td>
<td>Reference Partitioning</td>
<td>F1T737P0</td>
<td>F1_REMOTE_MSG_ID, SEQNO</td>
<td>Global</td>
<td>RANGE(F1_REMOTE_MSG_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T737S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1T737S2</td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_REMOTE_MSG_LOG</td>
<td>Reference Partitioning</td>
<td>F1T738P0</td>
<td>F1_REMOTE_MSG_ID, SEQNO, PARM_SEQ</td>
<td>Global</td>
<td>RANGE(F1_REMOTE_MSG_ID)</td>
<td></td>
</tr>
<tr>
<td>F1_REMOTE_MSG_LOG_PARM</td>
<td>Reference Partitioning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Statistics Snapshot

This table describes the Statistics Snapshot maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_STATS_SNPSHT(Parent)</td>
<td>RANGE (ILM_DT, SNAPSHOT_ID)</td>
<td>F1C706P0</td>
<td>SNAPSHOT_ID</td>
<td>Global</td>
<td>RANGE (SNAPSHOT_ID)</td>
<td>F1_STATS_SNPSHT_CRE_DTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C706S1</td>
<td>BUS_OBJ_CD, BO_STATUS_CD, SNAPSHOT_ID</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CM_ILM_F1C706S2</td>
<td>ILM_DT, ILM_ARCH_SW, SNAPSHOT_ID</td>
<td>Local</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>F1_STATS_SNPSHT_CHAR</td>
<td>Reference Partitioning</td>
<td>F1C707P0</td>
<td>SNAPSHOT_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C707S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1_STATS_SNPSHT_LOG</td>
<td>Reference Partitioning</td>
<td>F1C708P0</td>
<td>SNAPSHOT_ID, SEQNO</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C708S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F1C708S2</td>
<td>SNAPSHOT_ID, SEQNO, PARM_SEQ</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>F1_STATS_SNPSHT_REL_OBJ</td>
<td>Reference Partitioning</td>
<td>F1C710P0</td>
<td>SNAPSHOT_ID, STATS_SNPSHT_REL_OBJ_TYPE_FLG, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
</tbody>
</table>
### Activity

If sub retention periods will be defined for this MO, then please follow the guidelines set forth in section Module Specific ILM Implementation Details For Sub Retention.

This table describes the Activity maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_ACTIVITY (Parent)</td>
<td>RANGE (ILM_DT, D1_ACTIVITY_ID)</td>
<td>D1T319P0</td>
<td>D1_ACTIVITY_ID</td>
<td>Global Partitioned</td>
<td>RANGE (D1_ACTIVITY_ID)</td>
<td>D1_ACTIVITY_CRE_DTTM</td>
</tr>
<tr>
<td>Note: Default is to use sub-retention or use RANGE (ILM_DT, D1_ACTIVITY_ID) if not using sub-retention.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T319S0</td>
<td>BUS_OBJ_CD, BO_STATUS_CD, D1_ACTIVITY_ID</td>
<td>Global Partitioned</td>
<td>HASH (BUS_OBJ_CD, BO_STATUS_CD, D1_ACTIVITY_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CM_ILM_D1T319S1</td>
<td>ILM_DT, ILM_ARCH_SW, D1_ACTIVITY_ID</td>
<td>Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T320P0</td>
<td>D1_ACTIVITY_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global Partitioned</td>
<td>RANGE (D1_ACTIVITY_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T320S0</td>
<td>SRCH_CHAR_VAL</td>
<td>Global Partitioned</td>
<td>HASH (SRCH_CHAR_VAL)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T330P0</td>
<td>D1_ACTIVITY_ID, ACTIVITY_ID_TYPE_FLG</td>
<td>Global Partitioned</td>
<td>RANGE (D1_ACTIVITY_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T330S0</td>
<td>ACTIVITY_ID_TYPE_FLG, ID_VALUE</td>
<td>Global Partitioned</td>
<td>HASH (ACTIVITY_ID_TYPE_FLG, ID_VALUE)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T330S1</td>
<td>ACTIVITY_ID_TYPE_FLG, UPPER (ID_VALUE)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T321P0</td>
<td>D1_ACTIVITY_ID, SEQNO</td>
<td>Global Partitioned</td>
<td>RANGE (D1_ACTIVITY_ID)</td>
<td></td>
</tr>
</tbody>
</table>
### Communication In

This table describes the Communication In maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type</th>
<th>Global or Local</th>
<th>Index Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_COMM_IN (Parent)</td>
<td>RANGE(ILM_DT, D1_COMM_ID)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table:**

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type</th>
<th>Global or Local</th>
<th>Index Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_COMM_IN</td>
<td>RANGE(ILM_DT, D1_COMM_ID)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table Name</td>
<td>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</td>
<td>Index Name</td>
<td>Index Columns</td>
<td>Index Type</td>
<td>Global or Local</td>
<td>Index Partitioning Sub-Partitioning Key</td>
<td>ILM DT Initial Load</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------</td>
<td>----------------------------</td>
<td>--------------------------------</td>
<td>------------</td>
<td>-----------------</td>
<td>----------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>D1T386P0</td>
<td></td>
<td>D1_COMM_ID</td>
<td></td>
<td>Global</td>
<td>Partitioned</td>
<td>(D1_COMM_ID)</td>
<td></td>
</tr>
<tr>
<td>D1T386S1</td>
<td></td>
<td>BUS_OBJ_CD,</td>
<td></td>
<td>Global</td>
<td>Partitioned</td>
<td>(BUS_OBJ_CD, D1_COMM_ID)</td>
<td></td>
</tr>
<tr>
<td>CM_ILM_D1T386S1</td>
<td></td>
<td>ILM_DT, ILM_ARCH_SW,</td>
<td></td>
<td>Local</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T387P0</td>
<td></td>
<td>D1_COMM_ID, CHAR_TYPE_CD,</td>
<td></td>
<td>Global</td>
<td>Partitioned</td>
<td>(D1_COMM_ID)</td>
<td></td>
</tr>
<tr>
<td>D1T387S0</td>
<td></td>
<td>SRCH_CHAR_VAL</td>
<td></td>
<td>Global</td>
<td>Partitioned</td>
<td>(SRCH_CHAR_VAL)</td>
<td></td>
</tr>
<tr>
<td>D1T391P0</td>
<td></td>
<td>D1_COMM_ID, COMM_ID_TYPE_FLG</td>
<td></td>
<td>Global</td>
<td>Partitioned</td>
<td>(D1_COMM_ID)</td>
<td></td>
</tr>
<tr>
<td>D1T391S0</td>
<td></td>
<td>COMM_ID_TYPE_FLG, ID_VALUE</td>
<td></td>
<td>Global</td>
<td>Partitioned</td>
<td>(COMM_ID_TYPE_FLG, ID_VALUE)</td>
<td></td>
</tr>
<tr>
<td>D1T391S1</td>
<td></td>
<td>COMM_ID_TYPE_FLG, UPPER(ID_VALUE)</td>
<td></td>
<td>Global</td>
<td>Partitioned</td>
<td>(COMM_ID_TYPE_FLG, UPPER(ID_VALUE))</td>
<td></td>
</tr>
<tr>
<td>D1T388P0</td>
<td></td>
<td>D1_COMM_ID, SEQNO</td>
<td></td>
<td>Global</td>
<td>Partitioned</td>
<td>(D1_COMM_ID)</td>
<td></td>
</tr>
<tr>
<td>D1T388S1</td>
<td></td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td></td>
<td>Global</td>
<td>Partitioned</td>
<td>(CHAR_TYPE_CD, CHAR_VAL_FK1)</td>
<td></td>
</tr>
<tr>
<td>D1T388S2</td>
<td></td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td></td>
<td>Global</td>
<td>Partitioned</td>
<td>(CHAR_TYPE_CD, CHAR_VAL)</td>
<td></td>
</tr>
<tr>
<td>D1T389P0</td>
<td></td>
<td>D1_COMM_ID, SEQNO</td>
<td></td>
<td>Global</td>
<td>Partitioned</td>
<td>(D1_COMM_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Communication Out

This table describes the Communication Out maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_COMM_OUT</td>
<td>RANGE(ILM_DT, D1_COMM_ID)</td>
<td>D1T380P0</td>
<td>D1_COMM_ID, MAINT_OBJ_CD, COMM_REL_OBJ_TYPE_FLG</td>
<td>Global Partitioned</td>
<td>RANGE(D1_COMM_ID)</td>
<td>D1_COMM_OUT, CRE_DTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T380S1</td>
<td>BUS_OBJ_CD, BO_STATUS_CD, D1_COMM_ID</td>
<td>Global Partitioned</td>
<td>HASH(BUS_OBJ_CD, BO_STATUS_CD, D1_COMM_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CM_ILM_D1T380S1</td>
<td>ILM_DT, ILM_ARCH_SW, D1_COMM_ID</td>
<td>Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_COMM_OUT_CHAR</td>
<td>REFERENCE (D1_COMM_OUT_CHAR_FK)</td>
<td>D1T381P0</td>
<td>D1_COMM_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global Partitioned</td>
<td>RANGE(D1_COMM_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T381S0</td>
<td>SRCH_CHAR_VAL</td>
<td>Global Partitioned</td>
<td>HASH(SRCH_CHAR_VAL)</td>
<td></td>
</tr>
<tr>
<td>Table Name</td>
<td>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</td>
<td>Index Name</td>
<td>Index Columns</td>
<td>Index Type Global or Local</td>
<td>Index Partitioning Sub-Partitioning Key</td>
<td>ILM_DT Initial Load</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------</td>
<td>------------</td>
<td>---------------</td>
<td>-----------------------------</td>
<td>-----------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>D1_COMM_OUT_IDENTIFIER</td>
<td>REFERENCE (D1_COMM_OUT_IDENTIFIER_FK)</td>
<td>D1T385P0</td>
<td>D1_COMM_ID, COMM_ID_TYPE_FLG</td>
<td>Global Partitioned</td>
<td>RANGE(D1_COMM_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T385S0</td>
<td>COMM_ID_TYPE_FLG, ID_VALUE</td>
<td>Global Partitioned</td>
<td>HASH(COMM_ID_TYPE_FLG, ID_VALUE)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T385S1</td>
<td>COMM_ID_TYPE_FLG, UPPER(ID_VALUE)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_COMM_OUT_LOG</td>
<td>REFERENCE (D1_COMM_OUT_LOG_FK)</td>
<td>D1T382P0</td>
<td>D1_COMM_ID, SEQNO</td>
<td>Global Partitioned</td>
<td>RANGE(D1_COMM_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T382S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td>Global Partitioned</td>
<td>HASH(CHAR_TYPE_CD, CHAR_VAL_FK1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T382S2</td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td>Global Partitioned</td>
<td>HASH(CHAR_TYPE_CD, CHAR_VAL)</td>
<td></td>
</tr>
<tr>
<td>D1_COMM_OUT_LOG_PARM</td>
<td>REFERENCE (D1_COMM_OUT_LOG_PARM_FK)</td>
<td>D1T383P0</td>
<td>D1_COMM_ID, SEQNO, PARM_SEQ</td>
<td>Global Partitioned</td>
<td>RANGE(D1_COMM_ID)</td>
<td></td>
</tr>
<tr>
<td>D1_COMM_OUT_REL_OBJ</td>
<td>REFERENCE (D1_COMM_OUT_REL_OBJ_FK)</td>
<td>D1T384P0</td>
<td>D1_COMM_ID, MAINT_OBJ_CD, COMM_REL_OBJ_TYPE_FLG</td>
<td>Global Partitioned</td>
<td>RANGE(D1_COMM_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T384S0</td>
<td>PK_VALUE1, PK_VALUE2, PK_VALUE3, PK_VALUE4, PK_VALUE5, MAINT_OBJ_CD</td>
<td>Global Partitioned</td>
<td>HASH(PK_VALUE1, PK_VALUE2, PK_VALUE3, PK_VALUE4)</td>
<td></td>
</tr>
</tbody>
</table>
Device Event
If sub retention periods will be defined for this MO, then please follow the guidelines set forth in section Module Specific ILM Implementation Details For Sub Retention.

This table describes the Device Event maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_DVC_EVT (Parent)</td>
<td>RANGE(ILM_DT, DVC_EVT_ID)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D1_DVC_EVT_CREAT_ID</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Default is to use sub-retention or use RANGE (ILM_DT,DVC_EVT_T_ID) if not using sub-retention.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T400P0</td>
<td>DVC_EVT_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>RANGE(DVC_EVT_ID)</td>
<td>D1_DVC_EVT_CREAT_ID</td>
<td></td>
</tr>
<tr>
<td>D1T400S1</td>
<td>BUS_OBJ_CD, BO_STATUS_CD, DVC_EVT_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>HASH(BUS_OBJ_CD, BO_STATUS_CD, DVC_EVT_ID)</td>
<td>D1_DVC_EVT_CREAT_ID</td>
<td></td>
</tr>
<tr>
<td>D1T400S2</td>
<td>D1_DEVICE_ID, DVC_EVT_DTTM</td>
<td>Global</td>
<td>Partitioned</td>
<td>HASH(D1_DEVICE_ID, DVC_EVT_DTTM)</td>
<td>D1_DVC_EVT_CREAT_ID</td>
<td></td>
</tr>
<tr>
<td>D1T400S3</td>
<td>BUS_OBJ_CD, BO_STATUS_CD, D1_DEVICE_ID, DVC_EVT_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>HASH(BUS_OBJ_CD, BO_STATUS_CD, D1_DEVICE_ID, DVC_EVT_ID)</td>
<td>D1_DVC_EVT_CREAT_ID</td>
<td></td>
</tr>
<tr>
<td>CM_JLM_D1T400S4</td>
<td>ILM_DT, ILM_ARCH_SW, DVC_EVT_ID</td>
<td>Local</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_DVC_EVT_CHAR</td>
<td>REFERENCE (D1_DVC_EVT_CHAR_FK)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T401P0</td>
<td>DVC_EVT_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td>RANGE(DVC_EVT_ID)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T401S0</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td>Partitioned</td>
<td>HASH(SRCH_CHAR_VAL)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_DVC_EVT_IDENTIFIER</td>
<td>REFERENCE (D1_DVC_EVT_IDENTIFIER_FK)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T405P0</td>
<td>DVC_EVT_ID, DVC_EVT_ID_TYPE_FLG</td>
<td>Global</td>
<td>Partitioned</td>
<td>RANGE(DVC_EVT_ID)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T405S0</td>
<td>DVC_EVT_ID_TYPE_FLG, ID_VALUE</td>
<td>Global</td>
<td>Partitioned</td>
<td>HASH(DVC_EVT_ID_TYPE_FLG, ID_VALUE)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T405S1</td>
<td>DVC_EVT_ID_TYPE_FLG, UPPER(ID_VALUE)</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Completion Event

This table describes the Completion Event maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_COMPL_EVT</td>
<td>RANGE(ILM_DT, COMPL_EVT_ID)</td>
<td>D1T340P0</td>
<td>COMPL_EVT_ID</td>
<td>Global</td>
<td>RANGE(COMPL_EVT_ID)</td>
<td>D1_COMPL_EVT_CRF_DTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T402P0</td>
<td>DVC_EVT_ID, SEQNO</td>
<td>Global</td>
<td>RANGE(DVC_EVT_ID)</td>
<td></td>
</tr>
<tr>
<td>Table Name</td>
<td>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</td>
<td>Index Name</td>
<td>Index Columns</td>
<td>Index Type</td>
<td>Partitioning Sub-Partitioning Key</td>
<td>ILM_DT Initial Load</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------------------------------------</td>
<td>------------</td>
<td>---------------</td>
<td>------------</td>
<td>-----------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>D1T34080</td>
<td>D1_ACTIVITY_ID Global Partitioned HASH(D1_ACTIVITY_ID)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_</td>
<td>ILM_DT, ILM_ARCH_SW, DVC_EVT_ID Local</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T340S1</td>
<td>D1_ACTIVITY_ID Global Partitioned HASH(D1_ACTIVITY_ID)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T341P0</td>
<td>COMPL_EVT_ID, CHAR_TYPE_CD, SEQ_NUM Global Partitioned RANGE(COMPL_EVT_ID)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T341S1</td>
<td>SRCH_CHAR_VAL Global Partitioned HASH(SRCH_CHAR_VAL)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T342P0</td>
<td>COMPL_EVT_ID, SEQNO Global Partitioned RANGE(COMPL_EVT_ID)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T342S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK Global Partitioned RANGE(COMPL_EVT_ID)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T342S2</td>
<td>CHAR_TYPE_CD, CHAR_VAL Global Partitioned RANGE(COMPL_EVT_ID)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T343P0</td>
<td>COMPL_EVT_ID, SEQNO PARM_SEQ Global Partitioned RANGE(COMPL_EVT_ID)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T344P0</td>
<td>COMPL_EVT_ID, MAINT_OBJ_CD, COMPL_EVT_REL_OBJ_TYP_FLG Global Partitioned RANGE(COMPL_EVT_ID)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T344S0</td>
<td>PK_VALUE1, PK_VALUE2, PK_VALUE3, PK_VALUE4, PK_VALUE5, MAINT_OBJ_CD Global Partitioned HASH(PK_VALUE1, PK_VALUE2, PK_VALUE3, PK_VALUE4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Initial Measurement Data

If sub retention periods will be defined for this MO, then please follow the guidelines set forth in section Module Specific ILM Implementation Details For Sub Retention.

This table describes the Initial Measurement Data maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_INIT_MSRM T_DATA (Parent)</td>
<td>RANGE (ILM_DT,MEASR_COMP_ID)</td>
<td>D1T304P0</td>
<td>INIT_MSRMT_DATA_ID</td>
<td>Global Partitioned</td>
<td>RANGE (INIT_MSRMT_DATA_ID)</td>
<td>D1_INIT_MSRM T_DATA, CRE_DTTM</td>
</tr>
<tr>
<td>D1_INIT_MSRM T_DATA_CHAR</td>
<td>REFERENCE (D1_INIT_MSRMT_DATA_CHAR_FK)</td>
<td>D1T305P0</td>
<td>INIT_MSRMT_DATA_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global Partitioned</td>
<td>RANGE(INIT_MSRMT_DATA_ID)</td>
<td></td>
</tr>
<tr>
<td>D1_INIT_MSRM T_DATA_LOG</td>
<td>REFERENCE (D1_INIT_MSRMT_DATA_LOG_FK)</td>
<td>D1T306P0</td>
<td>INIT_MSRMT_DATA_ID, SEQNO</td>
<td>Global Partitioned</td>
<td>RANGE(INIT_MSRMT_DATA_ID)</td>
<td></td>
</tr>
<tr>
<td>D1_INIT_MSRM T_DATA_LOG_PARM</td>
<td>REFERENCE (D1_INIT_MSRMT_DATA_LOG_PARM_FK)</td>
<td>D1T305S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global Partitioned</td>
<td>HASH(SRCH_CHAR_VAL)</td>
<td></td>
</tr>
<tr>
<td>D1_INIT_MSRM T_DATA_L</td>
<td>REFERENCE (D1_INIT_MSRMT_DATA_L_FK)</td>
<td>D1T305S2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_INIT_MSRM T_DATA_LOG_PARM</td>
<td>REFERENCE (D1_INIT_MSRMT_DATA_LOG_PARM_FK)</td>
<td>D1T305S3</td>
<td>IMD_EXT_ID, INIT_MSRMT_DATA_ID</td>
<td>Global Partitioned</td>
<td>HASH(IMD_EXT_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T305S4</td>
<td>IMD_EXT_ID, ILM_DT, ILM_ARCH_SW, INIT_MSRMT_DATA_ID</td>
<td>Local</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Usage Transaction

This table describes the Usage Transaction maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_USAGE</td>
<td>RANGE(ILM_DT,  D1_USAGE_ID)</td>
<td>D1T281P0</td>
<td>D1_USAGE_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td>D1_USAGE_CRE_DTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T281S0</td>
<td>US_ID, START_DTTM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T281S1</td>
<td>BUS_OBJ_CD,  BO_STATUS_CD, D1_USAGE_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CM_ILM_</td>
<td>ILM_DT,  ILM_ARCH_SW, D1_USAGE_ID</td>
<td>D1T281S2</td>
<td>ILM_DT</td>
<td>Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_USAGE_</td>
<td>REFERENCE (D1_USAGE_, CHAR_FK)</td>
<td>D1T419S1</td>
<td>USG_EXT_ID, D1_USAGE_ID</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>CHAR</td>
<td></td>
<td>D1T285P0</td>
<td>D1_USAGE_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td>D1_USAGE_LOG</td>
<td>REFERENCE (D1_USAGE_LOG _FK)</td>
<td>D1T285S1</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T286P0</td>
<td>D1_USAGE_ID, SEQNO</td>
<td>Global</td>
<td>Partitioned</td>
<td></td>
</tr>
</tbody>
</table>

**Table:** Usage Transaction

**Usage Transaction**

This table describes the Usage Transaction maintenance object.
<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1T286S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td></td>
<td></td>
<td></td>
<td>Hash(CHAR_TYPE_CD, CHAR_VAL_FK1)</td>
<td></td>
</tr>
<tr>
<td>D1T286S2</td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td></td>
<td></td>
<td></td>
<td>Hash(CHAR_TYPE_CD, CHAR_VAL)</td>
<td></td>
</tr>
<tr>
<td>D1_USAGE_LOG_PARM</td>
<td>REFERENCE(D1_USAGE_LOG_PARM_FK)</td>
<td>D1T287P0</td>
<td>D1_USAGE_ID, SEQNO, PARM_SEQ</td>
<td>Global Partitioned</td>
<td>RANGE(D1_USAGE_ID)</td>
<td></td>
</tr>
<tr>
<td>D1_USAGE_PERIOD</td>
<td>REFERENCE(D1_USAGE_PERIOD_FK)</td>
<td>D1T283P0</td>
<td>D1_USAGE_ID, PERIOD_SEQ_NUM</td>
<td>Global Partitioned</td>
<td>RANGE(D1_USAGE_ID)</td>
<td></td>
</tr>
<tr>
<td>D1_USAGE_PERIOD_ITEM_DET</td>
<td>REFERENCE(D1_USAGE_PERIOD_ITEM_DET_FK)</td>
<td>D1T431P0</td>
<td>D1_USAGE_ID, PERIOD_SEQ_NUM, ITEM_SEQ_NUM</td>
<td>Global Partitioned</td>
<td>RANGE(D1_USAGE_ID)</td>
<td></td>
</tr>
<tr>
<td>D1_USAGE_PERIOD_SQ</td>
<td>REFERENCE(D1_USAGE_PERIOD_SQ_FK)</td>
<td>D1T284P0</td>
<td>D1_USAGE_ID, PERIOD_SEQ_NUM, SQ_SEQ_NUM</td>
<td>Global Partitioned</td>
<td>RANGE(D1_USAGE_ID)</td>
<td></td>
</tr>
<tr>
<td>D1_USAGE_PERIOD_SQ_DATA</td>
<td>REFERENCE(D1_USAGE_PERIOD_SQ_DATA_FK)</td>
<td>D1T497P0</td>
<td>D1_USAGE_ID, PERIOD_SEQ_NUM, SQ_SEQ_NUM, SQ_DATA_DTTM</td>
<td>Global Partitioned</td>
<td>RANGE(D1_USAGE_ID)</td>
<td></td>
</tr>
<tr>
<td>D1_USAGE_REL</td>
<td>REFERENCE(D1_USAGE_REL_FK)</td>
<td>D1T316P0</td>
<td>D1_USAGE_ID, USAGE_REL_TYPE_FLG</td>
<td>Global Partitioned</td>
<td>RANGE(D1_USAGE_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T316S0</td>
<td>REL_USAGE_ID, USAGE_REL_TYPE_FLG, D1_USAGE_ID</td>
<td>Global Partitioned</td>
<td>Hash(REL_USAGE_ID, USAGE_REL_TYPE_FLG, D1_USAGE_ID)</td>
<td></td>
</tr>
</tbody>
</table>
Usage Transaction Exception
This table describes the Usage Transaction Exception maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_USAGE_EXCP</td>
<td>REFERENCE(D1_USAGE_EXCP_CHAR_FK)</td>
<td>D1T443P0</td>
<td>Global</td>
<td>RANGE(USAGE_EXCP_ID)</td>
<td>D1_USAGE_EXCP_CRE_DTTM</td>
</tr>
<tr>
<td>D1_USAGE_EXCP_CHAR</td>
<td>REFERENCE(D1_USAGE_EXCP_CHAR_FK)</td>
<td>D1T446P0</td>
<td>Global</td>
<td>RANGE(USAGE_EXCP_ID)</td>
<td></td>
</tr>
<tr>
<td>D1_USAGE_EXCP_PARM</td>
<td>REFERENCE(D1_USAGE_EXCP_PARM_FK)</td>
<td>D1T445P0</td>
<td>Global</td>
<td>RANGE(USAGE_EXCP_ID)</td>
<td></td>
</tr>
</tbody>
</table>
**VEE Exception**

This table describes the VEE Exception maintenance object.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_VEE_EXCP</td>
<td>RANGE(ILM_DT, VEE_EXCP_ID)</td>
<td>D1T308P0</td>
<td>VEE_EXCP_ID</td>
<td>Global</td>
<td>RANGE(VEE_EXCP_ID)</td>
<td>D1_VEE_EXCP.CRE_DTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T308S1</td>
<td>INIT_MSRMT_DATA_ID</td>
<td>Global</td>
<td>HASH(INIT_MSRMT_DATA_ID)</td>
<td></td>
</tr>
<tr>
<td>CM_ILM_</td>
<td></td>
<td>CM_T308S2</td>
<td>ILM_DT,</td>
<td>Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILM_ARCH_SW,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VEE_EXCP_ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_VEE_EXCP_CHAR</td>
<td>REFERENCE (D1_VEE_EXCP_CHAR_FK)</td>
<td>D1T310P0</td>
<td>VEE_EXCP_ID,</td>
<td>Global</td>
<td>RANGE(VEE_EXCP_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CHAR_TYPE_CD,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SEQ_NUM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_VEE_EXCP_PARM</td>
<td>REFERENCE (D1_VEE_EXCP_PARM_FK)</td>
<td>D1T309P0</td>
<td>VEE_EXCP_ID,</td>
<td>Global</td>
<td>RANGE(VEE_EXCP_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PARM_SEQ</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Snapshot Tables**

This table below describes the snapshot tables.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_SNAPSHOT_DL_CTRL</td>
<td>RANGE(ILM_DT)</td>
<td>D1T433P0</td>
<td>SNAPSHOT_FACT_NAME_CD</td>
<td>Global</td>
<td>Partitioned</td>
<td>D1_SNAPSHOT_DL_CTRL_SNA_PSHOT_DTTM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_SNAPSHOT_DL_CTRL</td>
<td>RANGE(ILM_DT)</td>
<td>D1T433P0</td>
<td>SNAPSHOT_FACT_NAME_CD</td>
<td>Global</td>
<td>Partitioned</td>
<td>D1_SNAPSHOT_DL_CTRL_SNA_PSHOT_DTTM</td>
</tr>
<tr>
<td>Table Name</td>
<td>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</td>
<td>Index Name</td>
<td>Index Columns</td>
<td>Index Type Global or Local</td>
<td>Index Partitioning Sub-Partitioning Key</td>
<td>ILM_DT Initial Load</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------</td>
<td>------------</td>
<td>---------------</td>
<td>---------------------------</td>
<td>----------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>CM_ILM_D1T433S1</td>
<td>ILM_DT, ILM_ARCH_SW, SNAPSHOT_FACT_NAME_CD, SNAPSHOT_DTTM</td>
<td>Local</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_SP_SNAP_DL</td>
<td>RANGE(ILM_DT, SP_SNAP_ID)</td>
<td>D1T434P0</td>
<td>SP_SNAP_ID</td>
<td>Global Partitioned</td>
<td>RANGE(SP_SNAP_ID)</td>
<td>D1_SP_SNAP_DL</td>
</tr>
<tr>
<td>CM_ILM_D1T434S0</td>
<td>ILM_DT, ILM_ARCH_SW, SP_SNAP_ID</td>
<td>Local</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_SP_ID, SNAPSHOT_DTTM, SNAPSHOT_TYPE_FLG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_D1T434S1</td>
<td>ILM_DT, ILM_ARCH_SW, SP_SNAP_ID</td>
<td>Local</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_SP_UNR_USG_SNAP_DL</td>
<td>RANGE(ILM_DT, SP_UNR_USG_SNAP_ID)</td>
<td>D1T438P0</td>
<td>SP_UNR_USG_SNAP_ID</td>
<td>Global Partitioned</td>
<td>RANGE(SP_UNR_USG_SNAP_ID)</td>
<td>D1_SP_UNR_USG_SNAP_DL</td>
</tr>
<tr>
<td>D1_SP_ID, SNAPSHOT_DTTM, MEASR_COMP_ID, USG_SNAPSHOT_TYPE_FLG, D1_TOU_CD, MSRMT_COND_FLG, SNAPSHOT_TYPE_FLG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T436S0</td>
<td>D1_SP_ID, SNAPSHOT_DTTM, MEASR_COMP_ID, USG_SNAPSHOT_TYPE_FLG, D1_TOU_CD, MSRMT_COND_FLG, SNAPSHOT_TYPE_FLG</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM_ILM_D1T436S1</td>
<td>ILM_DT, ILM_ARCH_SW, SP_USG_SNAP_ID</td>
<td>Local</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_SP_VEE_EXCP_SNAP_DL</td>
<td>RANGE(ILM_DT, SP_VEE_EXCP_SNAP_ID)</td>
<td>D1T440P0</td>
<td>SP_VEE_EXCP_SNAP_ID</td>
<td>Global Partitioned</td>
<td>RANGE(SP_VEE_EXCP_SNAP_ID)</td>
<td>D1_SP_VEE_EXCP_SNAP_DL</td>
</tr>
<tr>
<td>D1_SP_ID, SNAPSHOT_DTTM, MEASR_COMP_ID, EXCP_TYPE_CD, D1_IMD_TYPE_FLG, EXCP_SEVERITY_FLG, VEE_GRP_CD, VEE_RULE_CD, SNAPSHOT_TYPE_FLG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1T440S0</td>
<td>D1_SP_ID, SNAPSHOT_DTTM, MEASR_COMP_ID, EXCP_TYPE_CD, D1_IMD_TYPE_FLG, EXCP_SEVERITY_FLG, VEE_GRP_CD, VEE_RULE_CD, SNAPSHOT_TYPE_FLG</td>
<td>Global Partitioned</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Information Lifecycle Management and MDM Data Archiving in C2M 7 - 30
Oracle Utilities Customer To Meter Database Administrator’s Guide
Module Specific ILM Implementation Details For Sub Retention

This section outlines each maintenance object that has been configured to support ILM as well as sub retention periods. This differs from the standard ILM enabled tables in that the partitioning strategy is inclusive of an additional column that defines the retention period for each record. In each case, the recommendation of the initial load of the ILM_DT and the <field name for retention period> for existing records is noted. The CTAS operation for these tables includes an extra step of generating a temporary mapping table that will allow the select for the ILM_DT to also identify the appropriate <retention period field name> for each record.

This section details the following maintenance objects that support ILM as well as sub retention periods:

- **Activity**
- **Device Event**
- **Initial Measurement Data**

### Activity

If sub retention periods will not be defined for this MO, then please follow the guidelines set forth in section Module Specific ILM Implementation Details.
<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_ACTIVITY_CHAR</td>
<td>REFERENCE (D1_ACTIVITY_CHAR_FK)</td>
<td>D1T320P0</td>
<td>D1_ACTIVITY_ID, CHAR_TYPE_CD, SEQ_NUM</td>
<td>Global Partitioned</td>
<td>RANGE(D1_ACTIVITY_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T320S0</td>
<td>SRCH_CHAR_VAL</td>
<td>Global Partitioned</td>
<td>HASH(SRCH_CHAR_VAL)</td>
<td></td>
</tr>
<tr>
<td>D1_ACTIVITY_IDENTIFIER</td>
<td>REFERENCE (D1_ACTIVITY_IDENTITY_FK)</td>
<td>D1T330P0</td>
<td>D1_ACTIVITY_ID, ACTIVITY_ID_TYPE_FLG</td>
<td>Global Partitioned</td>
<td>RANGE(D1_ACTIVITY_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T330S0</td>
<td>ACTIVITY_ID_TYPE_FLG, ID_VALUE</td>
<td>Global Partitioned</td>
<td>HASH(ACTIVITY_ID_TYPE_FLG, ID_VALUE)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T330S1</td>
<td>ACTIVITY_ID_TYPE, ID_VALUE</td>
<td>Global Partitioned</td>
<td>HASH(ID_VALUE)</td>
<td></td>
</tr>
<tr>
<td>D1_ACTIVITY_LOG</td>
<td>REFERENCE (D1_ACTIVITY_LOG_FK)</td>
<td>D1T321P0</td>
<td>D1_ACTIVITY_ID, SEQNO</td>
<td>Global Partitioned</td>
<td>RANGE(D1_ACTIVITY_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T321S1</td>
<td>CHAR_TYPE_CD, CHAR_VAL_FK1</td>
<td>Global Partitioned</td>
<td>HASH(CHAR_TYPE_CD, CHAR_VAL_FK1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T321S2</td>
<td>CHAR_TYPE_CD, CHAR_VAL</td>
<td>Global Partitioned</td>
<td>HASH(CHAR_TYPE_CD, CHAR_VAL)</td>
<td></td>
</tr>
<tr>
<td>D1_ACTIVITY_LOG_PARM</td>
<td>REFERENCE (D1_ACTIVITY_LOG_PARM_FK)</td>
<td>D1T322P0</td>
<td>D1_ACTIVITY_ID, SEQNO, PARM_SEQ</td>
<td>Global Partitioned</td>
<td>RANGE(D1_ACTIVITY_ID)</td>
<td></td>
</tr>
<tr>
<td>D1_ACTIVITY_REL</td>
<td>REFERENCE (D1_ACTIVITY_REL_FK)</td>
<td>D1T323P0</td>
<td>D1_ACTIVITY_ID, ACTIVITY_REL_TYPE_FLG</td>
<td>Global Partitioned</td>
<td>RANGE(D1_ACTIVITY_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T323S0</td>
<td>REL_ACTIVITY_ID</td>
<td>Global Partitioned</td>
<td>HASH(REL_ACTIVITY_ID)</td>
<td></td>
</tr>
</tbody>
</table>

Information Lifecycle Management and MDM Data Archiving in C2M 7 - 32
Oracle Utilities Customer To Meter Database Administrator's Guide
<table>
<thead>
<tr>
<th>Table Name</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_ACTIVITY_REL_OBJ</td>
<td>D1_activity__REL_OBJ</td>
<td>D1_ACTIVITY_ID, MAINT_OBJ_CD, ACTIVITY_REL_OBJ_TYPE_FLG</td>
<td>Global Partitioned</td>
<td>RANGE(D1_ACTIVITY_ID)</td>
<td></td>
</tr>
<tr>
<td>D1T324P0</td>
<td>D1T324P0</td>
<td>PK_VALUE1, PK_VALUE2, PK_VALUE3, PK_VALUE4, PK_VALUE5, MAINT_OBJ_CD</td>
<td>Global Partitioned</td>
<td>HASH(PK_VALUE1, PK_VALUE2, PK_VALUE3, PK_VALUE4)</td>
<td></td>
</tr>
<tr>
<td>D1T324S0</td>
<td>D1T324S0</td>
<td>PK_VALUE1, PK_VALUE2, PK_VALUE3, PK_VALUE4</td>
<td>Global Partitioned</td>
<td>HASH(PK_VALUE1, PK_VALUE2, PK_VALUE3, PK_VALUE4)</td>
<td></td>
</tr>
</tbody>
</table>
Query for Setting the Retention Period

The following query should be used to create a temporary table to create a mapping table that will identify the retention period for each measuring component type. This table will then be used during in the CTAS operation for Activity to identify the retention period for each record.

Please refer to Appendix B: Sample SQL for Enabling ILM with Sub Retention in MDM (Existing Installation) for detailed information using Initial Measurement Data as an example.

**Note:** A pre-requisite to executing this query is configuring the appropriate retention periods in the ILM master configuration in the Oracle Utilities Meter Data Management application.

```sql
/* *****ACTIVITY***** */
CREATE TABLE ILM_ACTIVITY_RETENTION_TMP
AS
  select acty.activity_type_cd
  /*retrieve the retention period for Activity Types in this order of precedence:
   1. The category based retention period from the MDM master configuration
   2. The MO level retention period from the MO options
   3. The installation level retention period from the FW master configuration*/
  , CAST(coalesce(catMap.retPeriod --Category level
  , (select maint_obj_opt_val
    from ci_md_mo_opt mmo
    where maint_obj_cd = 'D1-ACTIVITY'
    and maint_obj_opt_flg = 'FLRP'
    and seq_num =
    (select max(seq_num)
      from ci_md_mo_opt mmo
      where maint_obj_cd = 'D1-ACTIVITY'
      and maint_obj_opt_flg = 'FLRP')) --MO level
  , extractvalue(xmlparse(content fw_mcfg.mst_config_data)
  , 'generalMasterConfiguration/defaultRetentionPeriod') --Install level
  ) as NUMBER(5)) retPeriod
  from d1_activity_type acty
  , (select extractvalue(value(p),
    'activityTypeCategoryRetentionPeriodList/activityTypeCategory'
    )ACTIVITY_TYPE_CAT_FLG
  , extractvalue(value(p),
    'activityTypeCategoryRetentionPeriodList/retentionPeriod'
    )retPeriod
  )ACTIVITY_TYPE_CAT_FLG
  , (select extractvalue(value(p),
    'activityTypeCategoryRetentionPeriodList/activityTypeCategory'
    )ACTIVITY_TYPE_CAT_FLG
  , extractvalue(value(p),
    'activityTypeCategoryRetentionPeriodList/retentionPeriod'
    )retPeriod
  )ACTIVITY_TYPE_CAT_FLG
  , f1_mst_config mdm_mcfg,
  table(xmlsequence(extract(xmlparse(content mdm_mcfg.mst_config_data),
    'activityRetentionPeriod/activityTypeCategoryRetentionPeriods/
    activityTypeCategoryRetentionPeriodList'
    ))) p
  , f1_mst_config fw_mcfg
  , mdm_mcfg.bus_obj_cd = 'D1-ILMMSConfig')catMap
  , f1_mst_config fw_mcfg
  , fw_mcfg.bus_obj_cd = 'F1-ILMMSConfig'
  and acty.ACTIVITY_TYPE_CAT_FLG = catMap.ACTIVITY_TYPE_CAT_FLG (+)
order by 1;
```
**Device Event**

*Note:* If sub retention periods will not be defined for this MO, then please follow the guidelines set forth in section Module Specific ILM Implementation Details.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_DVC_EVT</td>
<td>RANGE(ILM_DT, RETENTION_PERIOD)</td>
<td>D1T400P0</td>
<td>DVC_EVT_ID</td>
<td>Global</td>
<td>RANGE</td>
<td>D1_DVC_EVT</td>
</tr>
<tr>
<td></td>
<td>(Parent)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CRE_DTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T400S1</td>
<td>BUS_OBJ_CD, BO_STATUS_CD,</td>
<td>Global</td>
<td>HASH(BUS_OBJ_CD, BO_STATUS_CD, DVC_EVT_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DVC_EVT_ID</td>
<td>Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T400S2</td>
<td>D1_DEVICE_ID, DVC_EVT_DTTM</td>
<td>Global</td>
<td>HASH(D1_DEVICE_ID, DVC_EVT_DTTM)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T400S3</td>
<td>BUS_OBJ_CD, BO_STATUS_CD,</td>
<td>Global</td>
<td>HASH(BUS_OBJ_CD, BO_STATUS_CD, DVC_EVT_ID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D1_DEVICE_ID, DVC_EVT_ID</td>
<td>Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CM_ILM_</td>
<td>ILM_DT, RETENTION_PERIOD,</td>
<td>Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>D1T400S4</td>
<td></td>
<td>ILM_ARCH_SW, DVC_EVT_ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_DVC_EVT_CHAR</td>
<td>REFERENCE</td>
<td>D1T401P0</td>
<td>DVC_EVT_ID, CHAR_TYPE_CD,</td>
<td>Global</td>
<td>RANGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(D1_DVC_EVT_CHAR_FK)</td>
<td></td>
<td>SEQ_NUM</td>
<td>Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_DVC_EVT_IDENTIFIER</td>
<td>REFERENCE</td>
<td>D1T401S0</td>
<td>SRCH_CHAR_VAL</td>
<td>Global</td>
<td>HASH(SRCH_CHAR_VAL)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(D1_DVC_EVT_IDENTIFIER_FK)</td>
<td></td>
<td></td>
<td>Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_DVC_EVT_LOG</td>
<td>REFERENCE</td>
<td>D1T405P0</td>
<td>DVC_EVT_ID, DVC_EVT_ID_TYPE_FLG</td>
<td>Global</td>
<td>RANGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(D1_DVC_EVT_LOG_FK)</td>
<td></td>
<td></td>
<td>Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T405S0</td>
<td>DVC_EVT_ID_TYPE_FLG, ID_VALUE</td>
<td>Global</td>
<td>HASH(DVC_EVT_ID_TYPE_FLG, ID_VALUE)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T405S1</td>
<td>DVC_EVT_ID_TYPE_FLG, UPPER(ID_VALUE)</td>
<td>Global</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Partitioned</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1T402P0</td>
<td>DVC_EVT_ID, SEQNO</td>
<td>Global</td>
<td>RANGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Partitioned</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The following query should be used to create a temporary table to create a mapping table that will identify the retention period for each measuring component type. This table will then be used during in the CTAS operation for Device Event to identify the retention period for each record.

Please refer to Appendix B: Sample SQL for Enabling ILM with Sub Retention in MDM (Existing Installation) for detailed information using Initial Measurement Data as an example.

**Note**: A pre-requisite to executing this query is configuring the appropriate retention periods in the ILM master configuration in the Oracle Utilities Meter Data Management application.

```sql
CREATE TABLE ILM_DVC_EVT_RETENTION_TMP
AS
SELECT det.dvc_evt_type_cd
/*retrieve the retention period for Device Event Types in this order of precedence:
  1. The category based retention period from the MDM master configuration
  2. The MO level retention period from the MO options
*/
```

---

**Query for Setting the Retention Period**

The following query should be used to create a temporary table to create a mapping table that will identify the retention period for each measuring component type. This table will then be used during in the CTAS operation for Device Event to identify the retention period for each record.

Please refer to Appendix B: Sample SQL for Enabling ILM with Sub Retention in MDM (Existing Installation) for detailed information using Initial Measurement Data as an example.

**Note**: A pre-requisite to executing this query is configuring the appropriate retention periods in the ILM master configuration in the Oracle Utilities Meter Data Management application.

```sql
CREATE TABLE ILM_DVC_EVT_RETENTION_TMP
AS
SELECT det.dvc_evt_type_cd
/*retrieve the retention period for Device Event Types in this order of precedence:
  1. The category based retention period from the MDM master configuration
  2. The MO level retention period from the MO options
*/
```
3. The installation level retention period from the FW master configuration

*/

    CAST(coalesce(catMap.retPeriod --Category level
    , (select maint_obj_opt_val
        from ci_md_mo_opt mmo
        where maint_obj_cd = 'D1-DVCEVENT'
        and maint_obj_opt_flg = 'FLRP'
        and seq_num = (select max(seq_num)
            from ci_md_mo_opt mmo
            where maint_obj_cd = 'D1-DVCEVENT'
            and maint_obj_opt_flg = 'FLRP')
    --MO level
    , extractvalue( xmlparse(content
        fw_mcfg.mst_config_data),
        'generalMasterConfiguration/defaultRetentionPeriod') --Install
level
) as NUMBER(5)) retPeriod
from d1_dvc_evt_type det
    , (select extractvalue(value(p),
        'deviceEventCategoryRetentionPeriodList/deviceEventCategory')
    dvc_evt_cat_flg
    , extractvalue(value(p),
        'deviceEventCategoryRetentionPeriodList/retentionPeriod')
    retPeriod
from f1_mst_config mdm_mcfg ,
    table(xmlsequence(extract(xmlparse(content
        mdm_mcfg.mst_config_data),
        'deviceEventRetentionPeriod/deviceEventCategoryRetentionPeriods/
        deviceEventCategoryRetentionPeriodList')) p
    where mdm_mcfg.bus_obj_cd = 'D1-ILMMSConfig')catMap
    , f1_mst_config fw_mcfg
where fw_mcfg.bus_obj_cd = 'F1-ILMMSConfig'
and det.dvc_evt_cat_flg = catMap.dvc_evt_cat_flg (+)
order by 1;

### Initial Measurement Data

If sub retention periods will not be defined for this MO, then please follow the guidelines set forth in section Module Specific ILM Implementation Details.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Partitioning Type (Partitioning, Sub-Partitioning Key)</th>
<th>Index Name</th>
<th>Index Columns</th>
<th>Index Type Global or Local</th>
<th>Index Partitioning Sub-Partitioning Key</th>
<th>ILM_DT Initial Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_INIT_MSRM T_DATA (Parent)</td>
<td>RANGE (ILM_DT, RETENTION_PERIOD)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D1_INIT_MSRM T_DATA</td>
</tr>
<tr>
<td>D1T304P0 INIT_MSRMT_DATA_ID</td>
<td>Global Partitioned</td>
<td>INIT_MSRMT_DATA_ID</td>
<td></td>
<td>RANGE (INIT_MSRMT_DATA_ID)</td>
<td></td>
<td>D1_INIT_MSRM T_DATA</td>
</tr>
<tr>
<td>D1T304S1 MEASR_COMP_ID, BO_STATUS_CD, BUS_OBJ_CD, D1_TO_DTTM, D1_FROM_DTTM</td>
<td>Global Partitioned</td>
<td>MEASR_COMP_ID, BO_STATUS_CD, BUS_OBJ_CD, D1_TO_DTTM, D1_FROM_DTTM</td>
<td></td>
<td>RANGE (MEASR_COMP_ID)</td>
<td></td>
<td>D1_INIT_MSRM T_DATA</td>
</tr>
</tbody>
</table>
Query for Setting the Retention Period

The following query should be used to create a temporary table to create a mapping table that will identify the retention period for each measuring component type. This table will then be used during in the CTAS operation for Initial Measurement Data to identify the retention period for each record.

Please refer to Appendix B: Sample SQL for Enabling ILM with Sub Retention in MDM (Existing Installation) for detailed information using Initial Measurement Data as an example.

**Note:** A pre-requisite to executing this query is configuring the appropriate retention periods in the ILM master configuration in the Oracle Utilities Meter Data Management application.

```
CREATE TABLE ILM_IMD_RETENTION_TMP
AS
select mct.measr_comp_type_cd
```
/* retrieve the retention period for MC Types in this order of precedence:
1. The UOM based retention period from the MDM master configuration
2. The interval IMD retention period from the MDM master configuration
3. The MO level retention period from the MO options
4. The installation level retention period from the FW master configuration */
,
CAST(coalesce( (select retPeriod
from (select 'D1IN' interval_scalar_flg,
extractvalue(value(p),'uomRetentionPeriodList/uom') D1_UOM_CD,
extractvalue(value(p),'uomRetentionPeriodList/retentionPeriod')
retPeriod
from f1_mst_config mdm_mcfg,
, table(xmlsequence(extract(xmlparse(content mdm_mcfg.mst_config_data),
'imdRetentionPeriod/intervalImdRetentionPeriods/uomRetentionPeriods/
retentionPeriodList')) p
where mdm_mcfg.bus_obj_cd = 'D1-ILMMSConfig'
union
select 'D1SC' INTERVAL_SCALAR_FLG,
extractvalue(value(p),'uomRetentionPeriodList/uom') D1_UOM_CD,
extractvalue(value(p),'uomRetentionPeriodList/retentionPeriod')
retPeriod
from f1_mst_config mdm_mcfg,
, table(xmlsequence(extract(xmlparse(content mdm_mcfg.mst_config_data),
'imdRetentionPeriod/scalarImdRetentionPeriods/uomRetentionPeriods/
retentionPeriodList')) p
where mdm_mcfg.bus_obj_cd = 'D1-ILMMSConfig') uomMap
where uomMap.interval_scalar_flg = mct.interval_scalar_flg
and trim(mctvi.d1_uom_cd) = trim(uomMap.d1_uom_cd))--UOM,
DECODE(mct.interval_scalar_flg
,'D1IN',
extractvalue( xmlparse(content mdm_mcfg.mst_config_data),
'imdRetentionPeriod/intervalImdRetentionPeriods/intervalRetentionPeriod') --interval IMD
,extractvalue( xmlparse(content mdm_mcfg.mst_config_data),
'imdRetentionPeriod/scalarImdRetentionPeriods/scalarRetentionPeriod')
--scalar IMD
)
, (select maint_obj_opt_val
from ci_md_mo_opt mno
where maint_obj_cd = 'D1-IMD'
and maint_obj_opt_flg = 'FLRP'
and seq_num = (select max(seq_num)
from ci_md_mo_opt mno
where maint_obj_cd = 'D1-IMD'
and maint_obj_opt_flg = 'FLRP') --IMD
, extractvalue( xmlparse(content fw_mcfg.mst_config_data),
'generalMasterConfiguration/defaultRetentionPeriod') --Install
) as NUMBER(5)) retPeriod
from d1_measr_comp_type mct
, d1_mc_type_value_identifier mctvi
, f1_mst_config fw_mcfg
, f1_mst_config mdm_mcfg
where mct.measr_comp_type_cd = mctvi.measr_comp_type_cd
and mctvi.value_id_type_flg = 'D1MS'
and fw_mcfg.bus_obj_cd = 'F1-ILMMSConfig'
and mdm_mcfg.bus_obj_cd = 'D1-ILMMSConfig'
order by 1;
On-going Maintenance Phase

The following steps provide a high level overview of what needs to be done for on-going maintenance for ILM on enabled MOs.

Please refer to the Appendix D: Sample SQL for Periodic Maintenance for detailed information using To Do Entry(Without LOB), F1_SYNC_REC_IN(With LOB-Tablespace per Partition), Initial Measurement Data (With LOB-Tablespace per Subpartition), and the D1_MSRMT table (Partition Compression) as examples.

1. Add the partition:
   a. Create Tablespace to be used for the new parent table partition.
   b. Since, we define MAXVALUE Partition; new partition can only be created using “SPLIT” operation. Identify and use next HIGHVALUE Partition for the split operation.
   c. All the child table(s) partition(s)LOB(s) must be altered to use the same tablespace as that of the parent table’s partition.
   d. Enable advanced compression on all child table(s).
   e. Copy partition level statistics from the previous partition.

2. Archive the partition/subpartition:
   a. Make the tablespace that will be archived READ ONLY.
   b. Check that no records have ILM_ARCH_SW = ‘N’.
      • If record count is zero, then proceed for further steps.
      • If record count is not zero, then change the tablespace back to READ WRITE MODE as Archive is not Feasible at the time.
   c. Create an archive tablespace for the partition/subpartition that needs to be archived.
   d. Create staging tables using the new archive tablespace. Load data for all child tables first.
   e. Create staging table using the new archive tablespace and load data for the parent table.
   f. Export tablespace using TRANSPORT_TABLESPACES method.
      Make Sure Tablespace datafile required for further import is preserved.
   g. Drop the partition, partition the tablespace and archive the tablespace (as it is already exported).

3. Restore the partition:
   a. Create a new tablespace to restore the partition/subpartition.
   b. Add partition using split operation on next greater high value partition.
      If the table contains LOBS, there will an additional statement in split partition DDL indicating tablespace where the LOBs will be stored.
   c. Enable advanced compression on all child table(s).
   d. Import Tablespace using TRANSPORT_TABLESPACES method.
e. Load data into the parent table first from the staging table.

f. Load data into the child table from the staging table.

g. Drop the archive tablespace after import and data loading is successful.

4. Compress D1_MSRMT table Partition:
   a. Create new compressed tablespace.
   b. Create a table using CTAS for each subpartition of the partition being compressed in the new compressed tablespace.
   c. Create a unique primary index for each subpartition of the partition being compressed in the new compressed tablespace. Then alter table to create the primary key.
   d. Exchange the subpartition of the D1_MSRMT table with the newly created table for each subpartition.
   e. Drop the original uncompressed tablespace.
   f. Alter the partition level metadata to reflect the new compressed tablespace.
   g. Rename the new compressed tablespace to the original tablespace name.

5. Move Data between different storage tiers:
The ILM facilities can be used within the database to implement storage savings, as follows:
   • 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.

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
**Naming Convention**

The ILM Assistant can provide the following:

- Setup ILM Lifecycle definition - Here you can define different lifecycle definitions for different MOs and configure when 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 so that when data is moved from one partition to another it will be automatically compressed to a desired degree.
- Lifecycle Management - There is a tab called Lifecycle Management where the system admin will be alerted when partitions are eligible for archiving.

ILM Assistant can then be used to ensure the records that have ILM_ARCH_SW = 'Y' can be archived or purged, as deemed appropriate by the business.

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

**Naming Convention**

The naming convention for tablespace, partitions & subpartition is standardized as follows:

- Each name consists of some or all of the following parts.
- The parts of the name are organized hierarchically.
- Each part of the Name is separated with an underscore.
- The maximum name length must not exceed 30 Characters.
- For an MO, the parent table and child table share the same tablespace for the corresponding partition (or sub partition as appropriate).
- Square brackets [ ] indicate that this part of the name should be omitted if not required.
OWNERFLAG_TABLEIDENTIFIER_PARTITIONNAME[_SUBPARTITIONNAME][_ARCHIVEFLAG][_COMPRESSFLAG]

For details on the convention, please refer to the table below:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNERFLAG</td>
<td>Owner flag for the relevant application for example “D1” for MDM</td>
</tr>
<tr>
<td>TABLE IDENTIFIER</td>
<td>The Index Name of the Primary Key index without the “P0” suffix. For example, if the PK index name is XT039P0, the table identifier would be “XT039”.</td>
</tr>
</tbody>
</table>
| PARTITION NAME      | The Partition name should be prefixed with a P followed by a name which conforms to one of the following standards:  
                     • 4 digit year and 3 letter month abbreviation PYYYYMON corresponding to the ILM date e.g. P2011JAN  
                     • PMAX if it is the Max Value partition |
| SUBPARTITION NAME   | If subpartitions are used, name should be prefixed with S followed by a name of not more than 5 characters which conforms to the following requirements:  
                     • SMAX if this is the Max Value sub partition  
                     • If the sub partition holds data for a sub retention period use a number equal to that period e.g S91 if the sub retention period < 91 days.  
                     • For a range based SubPartition on Primary Key, use an integral number increasing by +1. For example, if there are 8 sub partitions use S01 through S08 |
| ARCHIVEFLAG         | This flag is used as a suffix to the table and tablespace name for the staging tables created for the archiving operation.  
                     • ARC |
<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
</table>
| COMPRESS FLAG | This flag is used as a suffix to the tablespace name for the staging tables created when compressing a partition.  
  - C  
  For compression related tasks, this is used as suffix to the tablespace name.  
  - Partition Tablespace Name: It is formed by OWNERFLAG_TABLEIDENTIFIER_PARTITIONNAME  
    For example:  
    CM_D1T304_PMAX  
    CM_D1T304_P2011JAN  
  - SubPartition Tablespace Name: It is formed by OWNERFLAG_TABLEIDENTIFIER_PARTITIONNAME_SUBPARTITIONNAME  
    For example:  
    CM_D1T304_PMAX_SMAX  
    CM_D1T304_P2011JAN_SMAX  
    CM_D1T304_PMAX_S001  
    CM_D1T304_P2011JAN_S181  
  - Archive Staging Table And Its Tablespace Name (When archiving partition): It is formed by OWNERFLAG_TABLEIDENTIFIER_PARTITIONNAME_ARCHIVEFLAG.  
    For example:  
    CM_D1T304_P2011JAN_ARC  
  - Archive Staging Table And Its Tablespace Name (When archiving subpartition): It is formed by OWNERFLAG_TABLEIDENTIFIER_PARTITIONNAME_SUBPARTITIONNAME_ARCHIVEFLAG.  
    For example:  
    CM_D1T304_P2011JAN_S181_ARC  
  - Compressed Tablespace name (When compressing partition):  
    For example:  
    CM_D1T304_P2011JAN_C |
Appendix A
Sample SQL for Enabling ILM in C2M for CC&B (Initial Install)

This section provides more detail about steps needed to fully support ILM on tables for maintenance objects that support the functionality. Two maintenance objects are shown:

- To Do Entry, which does not include a LOB field.
- Sync Request, which does include a LOB field.

Other maintenance object's implementations can follow the appropriate pattern based on whether there is a LOB field or not.

The following DDL(s):

- Follow Naming convention recommendations for partitions\subpartitions\tablespaces.
- Ensure all the ILM Storage requirements are incorporated, failing which, ILM functionality will not be achieved.
  - Partitions are defined with respective Tablespace.
  - Child Tables are referenced partitioned.
- Ensure all compression recommendations are incorporated.

Maintenance Object: TO DO ENTRY

Parent Table: CI_TD_ENTRY

```
CREATE BIGFILE TABLESPACE CM_XT039_P2017JAN DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2017FEB DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2017MAR DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2017APR DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
```
CREATE BIGFILE TABLESPACE CM_XT039_P2017MAY DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2017JUN DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2017JUL DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2017AUG DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2017SEP DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2017OCT DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2017NOV DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2017DEC DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_PMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;

CREATE TABLE CI_TD_ENTRY (  
TD_ENTRY_ID     CHAR(14) NOT NULL ENABLE,  
BATCH_CD        CHAR(8) DEFAULT ' ' NOT NULL ENABLE,  
BATCH_NBR       NUMBER(10,0) DEFAULT 0 NOT NULL ENABLE,  
MESSAGE_CAT_NBR NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,  
MESSAGE_NBR     NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,  
ASSIGNED_TO     CHAR(8) DEFAULT ' ' NOT NULL ENABLE,  
TD_TYPE_CD      CHAR(8) DEFAULT ' ' NOT NULL ENABLE,  
ROLE_ID         CHAR(10) DEFAULT ' ' NOT NULL ENABLE,  
ENTRY_STATUS_FLG CHAR(2) DEFAULT ' ' NOT NULL ENABLE,  
VERSION         NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,  
CRE_DTTM DATE,  
ASSIGNED_DTTM DATE,  
COMPLETE_DTTM DATE,  
COMPLETE_USER_ID CHAR(8) DEFAULT ' ' NOT NULL ENABLE,  
COMMENTS        VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,  
ASSIGNED_USER_ID CHAR(8) DEFAULT ' ' NOT NULL ENABLE,  
TD_PRIORITY_FLG CHAR(4) DEFAULT ' ' NOT NULL ENABLE,  
ILM_DT DATE,  
ILM_ARCH_SW CHAR(1)  
)  
ENABLE ROW MOVEMENT  
PARTITION BY RANGE (ILM_DT)  
SUBPARTITION BY RANGE (TD_ENTRY_ID) SUBPARTITION TEMPLATE  
{  
SUBPARTITION S01 VALUES LESS THAN ( '124999999999' ),  
SUBPARTITION S02 VALUES LESS THAN ( '249999999999' ),  
SUBPARTITION S03 VALUES LESS THAN ( '374999999999' ),  
SUBPARTITION S04 VALUES LESS THAN ( '499999999999' ),  
SUBPARTITION S05 VALUES LESS THAN ( '624999999999' ),  
SUBPARTITION S06 VALUES LESS THAN ( '749999999999' ),  
SUBPARTITION S07 VALUES LESS THAN ( '874999999999' ),  
SUBPARTITION SMAX VALUES LESS THAN ( MAXVALUE )  
}  
}
PARTITION "P2017JAN" VALUES LESS THAN (TO_DATE('2017-02-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017JAN,
PARTITION "P2017FEB" VALUES LESS THAN (TO_DATE('2017-03-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017FEB,
PARTITION "P2017MAR" VALUES LESS THAN (TO_DATE('2017-04-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017MAR,
PARTITION "P2017APR" VALUES LESS THAN (TO_DATE('2017-05-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017APR,
PARTITION "P2017MAY" VALUES LESS THAN (TO_DATE('2017-06-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017MAY,
PARTITION "P2017JUN" VALUES LESS THAN (TO_DATE('2017-07-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017JUN,
PARTITION "P2017JUL" VALUES LESS THAN (TO_DATE('2017-08-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017JUL,
PARTITION "P2017AUG" VALUES LESS THAN (TO_DATE('2017-09-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017AUG,
PARTITION "P2017SEP" VALUES LESS THAN (TO_DATE('2017-10-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017SEP,
PARTITION "P2017OCT" VALUES LESS THAN (TO_DATE('2017-11-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017OCT,
PARTITION "P2017NOV" VALUES LESS THAN (TO_DATE('2017-12-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017NOV,
PARTITION "P2017DEC" VALUES LESS THAN (TO_DATE('2018-01-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017DEC,
PARTITION "PMAX" VALUES LESS THAN (MAXVALUE)
tablespace CM_XT039_PMAX
);

INDEX
CREATE BIGFILE TABLESPACE CM_XT039_IND DATAFILE '+DATADG' SIZE 50M
AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE UNIQUE INDEX XT039P0 ON CI_TD_ENTRY ( TD_ENTRY_ID ) TABLESPACE
CM_XT039_IND
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 ( '749999999999' ),
PARTITION P7 VALUES LESS THAN ( '874999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
};
ALTER TABLE CI_TD_ENTRY ADD CONSTRAINT XT039P0 PRIMARY
KEY(TD_ENTRY_ID) USING INDEX;
CREATE UNIQUE INDEX XT039S2 ON CI_TD_ENTRY (ASSIGNED_TO, TD_ENTRY_ID) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

CREATE INDEX XT039S3 ON CI_TD_ENTRY (ENTRY_STATUS_FLG, ASSIGNED_TO) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

CREATE INDEX XT039S4 ON CI_TD_ENTRY (ROLE_ID, TD_TYPE_CD, ENTRY_STATUS_FLG, CM_PRIORITY_FLG) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

CREATE INDEX XT039S5 ON CI_TD_ENTRY (BATCH_CD, BATCH_NBR, ENTRY_STATUS_FLG) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

CREATE UNIQUE INDEX XT039S6 ON CI_TD_ENTRY (TD_ENTRY_ID, ASSIGNED_TO) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

CREATE UNIQUE INDEX XT039S7 ON CI_TD_ENTRY (COMPLETE_USER_ID, COMPLETE_DTTM, TD_ENTRY_ID) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

CREATE UNIQUE INDEX ILM_XT039S8 ON CI_TD_ENTRY (ILM_DT, ILM_ARCH_SW, TD_ENTRY_ID) LOCAL COMPRESS ADVANCED LOW;

CREATE TABLE CI_TD_DRLKEY
(
    TD_ENTRY_ID CHAR(14) NOT NULL ENABLE,
    SEQ_NUM    NUMBER(3,0) NOT NULL ENABLE,
    KEY_VALUE  VARCHAR2(50) 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 ON DELETE CASCADE
) PARTITION BY REFERENCE (CI_TD_DRLKEY_FK) ENABLE ROW MOVEMENT;

INDEX

CREATE UNIQUE INDEX XT037P0 ON CI_TD_DRLKEY (TD_ENTRY_ID, SEQ_NUM) TABLESPACE CM_XT039_IND
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 ('749999999999'),
    PARTITION P7 VALUES LESS THAN ('874999999999'),
    PARTITION P8 VALUES LESS THAN (MAXVALUE)
)
COMPRESS ADVANCED LOW;

ALTER TABLE CI_TD_DRLKEY ADD CONSTRAINT XT037P0 PRIMARY KEY(TD_ENTRY_ID, SEQ_NUM) USING INDEX;
CREATE INDEX XT037S1 ON CI_TD_DRLKEY (KEY_VALUE, TD_ENTRY_ID)
TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

Child Table: CI_TD_ENTRY_CHA

CREATE TABLE CI_TD_ENTRY_CHA
(
    TD_ENTRY_ID CHAR(14) NOT NULL ENABLE,
    CHAR_TYPE_CD CHAR(8) NOT NULL ENABLE,
    SEQ_NUM NUMBER(3,0) DEFAULT 0 NOT NULL ENABLE,
    CHAR_VAL CHAR(16) DEFAULT ' ' NOT NULL ENABLE,
    VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    ADHOC_CHAR_VAL VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK1 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK2 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK3 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK4 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK5 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    SRCH_CHAR_VAL VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    CONSTRAINT CI_TD_ENTRY_CHA_FK FOREIGN KEY(TD_ENTRY_ID) REFERENCES
    CI_TD_ENTRY ON DELETE CASCADE
) PARTITION BY REFERENCE (CI_TD_ENTRY_CHA_FK)
ENABLE ROW MOVEMENT;

INDEX
CREATE UNIQUE INDEX XT701P0 ON CI_TD_ENTRY_CHA (TD_ENTRY_ID,
CHAR_TYPE_CD, SEQ_NUM) TABLESPACE CM_XT039_IND
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 ( '749999999999' ),
    PARTITION P7 VALUES LESS THAN ( '874999999999' ),
    PARTITION P8 VALUES LESS THAN ( MAXVALUE )
) COMPRESS ADVANCED LOW;
ALTER TABLE CI_TD_ENTRY_CHA ADD CONSTRAINT XT701P0 PRIMARY
KEY (TD_ENTRY_ID, CHAR_TYPE_CD, SEQ_NUM) USING INDEX;
CREATE INDEX XT701S1 ON CI_TD_ENTRY_CHA (SRCH_CHAR_VAL, CHAR_TYPE_CD,
TD_ENTRY_ID) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

Child Table: CI_TD_LOG

CREATE TABLE CI_TD_LOG
(
    TD_ENTRY_ID CHAR(14) NOT NULL ENABLE,
    SEQ_NUM NUMBER(3,0) NOT NULL ENABLE,
    LOG_DTTM DATE NOT NULL ENABLE,
    LOG_TYPE_FLG CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
 Sample SQL for Enabling ILM in C2M for CC&B (Initial Install) A - 6
Oracle Utilities Customer To Meter Database Administrator’s Guide

Child Table: CI_TD_MSG_PARM

CREATE TABLE CI_TD_MSG_PARM
(
TD_ENTRY_ID CHAR(14) NOT NULL ENABLE,
SEQ_NUM NUMBER(3,0) NOT NULL ENABLE,
MSG_PARM_VAL VARCHAR2(30) 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 ON DELETE CASCADE
)
PARTITION BY REFERENCE (CI_TD_MSG_PARM_FK)
ENABLE ROW MOVEMENT;

INDEX

CREATE UNIQUE INDEX XT040P0 ON CI_TD_MSG_PARM (TD_ENTRY_ID, SEQ_NUM) TABLESPACE CM_XT039_IND
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 ( '749999999999' ),
PARTITION P7 VALUES LESS THAN ( '874999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
)
COMPRESS ADVANCED LOW;

ALTER TABLE CI_TD_MSG_PARM ADD CONSTRAINT XT040P0 PRIMARY KEY(TD_ENTRY_ID, SEQ_NUM) USING INDEX;

CREATE INDEX XT041S1 ON CI_TD_MSG_PARM (LOG_DTTM, USER_ID, LOG_TYPE_FLG, TD_ENTRY_ID) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;
Child Table: CI_TD_SRTKEY

CREATE TABLE CI_TD_SRTKEY
(
  TD_ENTRY_ID CHAR(14) NOT NULL ENABLE,
  SEQ_NUM NUMBER(3,0) NOT NULL ENABLE,
  KEY_VALUE VARCHAR2(50) 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 ON DELETE CASCADE
)
PARTITION BY REFERENCE (CI_TD_SRTKEY_FK)
ENABLE ROW MOVEMENT;

INDEX
CREATE UNIQUE INDEX XT041P0 ON CI_TD_SRTKEY ( TD_ENTRY_ID, SEQ_NUM ) TABLESPACE CM_XT039_IND
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 ( '749999999999' ),
  PARTITION P7 VALUES LESS THAN ( '874999999999' ),
  PARTITION P8 VALUES LESS THAN ( MAXVALUE )
)
COMPRESS ADVANCED LOW;

ALTER TABLE CI_TD_SRTKEY ADD CONSTRAINT XT041P0 PRIMARY KEY(TD_ENTRY_ID, SEQ_NUM) USING INDEX;

CREATE INDEX XT041S1 ON CI_TD_SRTKEY ( KEY_VALUE, TD_ENTRY_ID ) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

Maintenance Object: F1-SYNCREQIN

Parent Table: F1_SYNC_REQ_IN

CREATE BIGFILE TABLESPACE CM_F1T191_P2017JAN DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2017FEB DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2017MAR DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2017APR DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2017MAY DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2017JUN DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2017JUL DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2017AUG DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2017SEP DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2017OCT DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2017NOV DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2017DEC DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_PMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;

CREATE TABLE F1_SYNC_REQ_IN
(
    F1_SYNC_REQ_IN_ID CHAR(14) NOT NULL ENABLE,
    BUS_OBJ_CD        CHAR(30) DEFAULT ' ' NOT NULL ENABLE,
    CRE_DTTM DATE NOT NULL ENABLE,
    BO_STATUS_CD CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
    STATUS_UPD_DTTM DATE,
    MAINT_OBJ_CD  CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
    NT_XID_CD     CHAR(30) DEFAULT ' ' NOT NULL ENABLE,
    EXT_PK_VALUE1 VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    EXT_PK_VALUE2 VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    EXT_PK_VALUE3 VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    EXT_PK_VALUE4 VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    EXT_PK_VALUE5 VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    PK_VALUE1     VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    BO_DATA_AREA CLOB,
    PRE_TRN_INIT_BO_DATA_AREA CLOB,
    PRE_TRN_FIN_BO_DATA_AREA CLOB,
    POST_TRN_BO_DATA_AREA CLOB,
    VERSION                  NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    EXT_REFERENCE_ID         CHAR(36) DEFAULT ' ' NOT NULL ENABLE,
    F1_INITIAL_LOAD_SYNC_FLG CHAR(14) DEFAULT ' ' NOT NULL ENABLE,
    F1_COMPOSITE_SYNC_FLG    CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
    ILM_DT DATE,
    ILM_ARCH_SW CHAR(1)
)
ENABLE ROW MOVEMENT
LOB (BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
LOB (PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
LOB (PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
LOB (POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
PARTITION BY RANGE(ILM_DT)
SUBPARTITION BY RANGE(F1_SYNC_REQ_IN_ID)
SUBPARTITION TEMPLATE
{
  SUBPARTITION S01 VALUES LESS THAN ("124999999999" ),
  SUBPARTITION S02 VALUES LESS THAN ("249999999999" ),
  SUBPARTITION S03 VALUES LESS THAN ("374999999999" ),
  SUBPARTITION S04 VALUES LESS THAN ("499999999999" ),
  SUBPARTITION S05 VALUES LESS THAN ("624999999999" ),
  SUBPARTITION S06 VALUES LESS THAN ("749999999999" ),
  SUBPARTITION S07 VALUES LESS THAN ("874999999999" ),
  SUBPARTITION SMAX VALUES LESS THAN (MAXVALUE )
}

PARTITION "P2017JAN" VALUES LESS THAN (TO_DATE(‘2017-02-01 00:00:01’,
’SYYYYY-MM-DD HH24:MI:SS’, ‘NLS_CALENDAR=GREGORIAN’))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017JAN )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017JAN )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017JAN )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE
tablespace CM_F1T191_P2017JAN )
tablespace CM_F1T191_P2017JAN,
PARTITION "P2017FEB" VALUES LESS THAN (TO_DATE(‘2017-03-01 00:00:01’,
’SYYYYY-MM-DD HH24:MI:SS’, ‘NLS_CALENDAR=GREGORIAN’))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017FEB )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017FEB )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017FEB )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017FEB )
tablespace CM_F1T191_P2017FEB,
PARTITION "P2017MAR" VALUES LESS THAN (TO_DATE(‘2017-04-01 00:00:01’,
’SYYYYY-MM-DD HH24:MI:SS’, ‘NLS_CALENDAR=GREGORIAN’))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017MAR )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017MAR )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017MAR )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017MAR )
tablespace CM_F1T191_P2017MAR,
PARTITION "P2017APR" VALUES LESS THAN (TO_DATE(‘2017-05-01 00:00:01’,
’SYYYYY-MM-DD HH24:MI:SS’, ‘NLS_CALENDAR=GREGORIAN’))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017APR )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017APR )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017APR )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017APR )
tablespace CM_F1T191_P2017APR,
PARTITION "P2017MAY" VALUES LESS THAN (TO_DATE('2017-06-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017MAY )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017MAY )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017MAY )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017MAY )
tablespace CM_F1T191_P2017MAY,
PARTITION "P2017JUN" VALUES LESS THAN (TO_DATE('2017-07-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017JUN )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017JUN )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017JUN )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017JUN )
tablespace CM_F1T191_P2017JUN,
PARTITION "P2017JUL" VALUES LESS THAN (TO_DATE('2017-08-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017JUL )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017JUL )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017JUL )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017JUL )
tablespace CM_F1T191_P2017JUL,
PARTITION "P2017AUG" VALUES LESS THAN (TO_DATE('2017-09-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017AUG )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017AUG )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017AUG )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017AUG )
tablespace CM_F1T191_P2017AUG,
PARTITION "P2017SEP" VALUES LESS THAN (TO_DATE('2017-10-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017SEP )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017SEP )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017SEP )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017SEP )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017SEP )
tablespace CM_F1T191_P2017SEP,
PARTITION "P2017OCT" VALUES LESS THAN (TO_DATE('2017-11-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017OCT )
LOB(PRE_TRN_INIT_ BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017OCT )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017OCT )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE
tablespace CM_F1T191_P2017OCT )
tablespace CM_F1T191_P2017OCT,
PARTITION "P2017NOV" VALUES LESS THAN (TO_DATE('2017-12-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017NOV )
LOB(PRE_TRN_INIT_ BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017NOV )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017NOV )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE
tablespace CM_F1T191_P2017NOV )
tablespace CM_F1T191_P2017NOV,
PARTITION "P2017DEC" VALUES LESS THAN (TO_DATE('2018-01-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017DEC )
LOB(PRE_TRN_INIT_ BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017DEC )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_P2017DEC )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE
tablespace CM_F1T191_P2017DEC )
tablespace CM_F1T191_P2017DEC,
PARTITION "PMAX" VALUES LESS THAN (MAXVALUE)
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_PMAX )
LOB(PRE_TRN_INIT_ BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_PMAX )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE tablespace CM_F1T191_PMAX )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE
tablespace CM_F1T191_PMAX )
tablespace CM_F1T191_PMAX
);

INDEX
CREATE BIGFILE TABLESPACE CM_F1T191_IND DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;

CREATE UNIQUE INDEX F1T191P0 ON F1_SYNC_REQ_IN(F1_SYNC_REQ_IN_ID)
TABLESPACE CM_F1T191_IND
GLOBAL PARTITION BY RANGE (F1_SYNC_REQ_IN_ID)
(
PARTITION P1 VALUES LESS THAN ( '124999999999' ),
PARTITION P2 VALUES LESS THAN ( '249999999999' ),

Sample SQL for Enabling ILM in C2M for CC&B (Initial Install) A - 11
Oracle Utilities Customer To Meter Database Administrator's Guide
PARTITION P3 VALUES LESS THAN ( '374999999999' ),
PARTITION P4 VALUES LESS THAN ( '499999999999' ),
PARTITION P5 VALUES LESS THAN ( '624999999999' ),
PARTITION P6 VALUES LESS THAN ( '749999999999' ),
PARTITION P7 VALUES LESS THAN ( '874999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE ) ;

ALTER TABLE F1_SYNC_REQ_IN ADD CONSTRAINT F1T191P0 PRIMARY KEY
(F1_SYNC_REQ_IN_ID) USING INDEX;

CREATE UNIQUE INDEX F1T191S1 ON F1_SYNC_REQ_IN (BO_STATUS_CD,
BUS_OBJ_CD, F1_SYNC_REQ_IN_ID) TABLESPACE CM_F1T191_IND COMPRESS
ADVANCED LOW;

CREATE INDEX F1T191S2 ON
F1_SYNC_REQ_IN(MAINT_OBJ_CD,EXT_PK_VALUE1,NT_XID_CD,PK_VALUE1)
TABLESPACE CM_F1T191_IND COMPRESS ADVANCED LOW;

CREATE UNIQUE INDEX CM_ILM_F1T191S3 ON F1_SYNC_REQ_IN(ILM_DT,
ILM_ARCH_SW, F1_SYNC_REQ_IN_ID) LOCAL COMPRESS ADVANCED LOW;

CREATE TABLE F1_SYNC_REQ_IN_CHAR
(
    F1_SYNC_REQ_IN_ID CHAR(14) NOT NULL ENABLE,
    CHAR_TYPE_CD    CHAR(8) NOT NULL ENABLE,
    SEQ_NUM         NUMBER(3,0) NOT NULL ENABLE,
    CHAR_VAL         CHAR(16) DEFAULT '' NOT NULL ENABLE,
    ADHOC_CHAR_VAL   VARCHAR2(254) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK1     VARCHAR2(50) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK2     VARCHAR2(50) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK3     VARCHAR2(50) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK4     VARCHAR2(50) DEFAULT '' NOT NULL ENABLE,
    CHAR_VAL_FK5     VARCHAR2(50) DEFAULT '' NOT NULL ENABLE,
    SRCH_CHAR_VAL    VARCHAR2(50) DEFAULT '' NOT NULL ENABLE,
    VERSION          NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    CONSTRAINT F1_SYNC_REQ_IN_CHAR_FK FOREIGN KEY(F1_SYNC_REQ_IN_ID)
    REFERENCES F1_SYNC_REQ_IN ON DELETE CASCADE
) PARTITION BY REFERENCE (F1_SYNC_REQ_IN_CHAR_FK)
ENABLE ROW MOVEMENT;

INDEX

CREATE UNIQUE INDEX F1T193P0 ON F1_SYNC_REQ_IN_CHAR(F1_SYNC_REQ_IN_ID,
CHAR_TYPE_CD, SEQ_NUM) TABLESPACE CM_F1T191_IND
GLOBAL PARTITION BY RANGE (F1_SYNC_REQ_IN_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 ( '749999999999' ),
    PARTITION P7 VALUES LESS THAN ( '874999999999' ),
    PARTITION P8 VALUES LESS THAN ( MAXVALUE )
}
COMPRESS ADVANCED LOW;

ALTER TABLE F1_SYNC_REQ_IN_CHAR ADD CONSTRAINT F1T193P0 PRIMARY KEY (F1_SYNC_REQ_IN_ID, CHAR_TYPE_CD, SEQ_NUM) USING INDEX;

CREATE INDEX F1T193S1 ON F1_SYNC_REQ_IN_CHAR(SRCH_CHAR_VAL) TABLESPACE CM_F1T191_IND ;

CREATE TABLE F1_SYNC_REQ_IN_EXCP

| F1_SYNC_REQ_IN_ID CHAR(14) NOT NULL ENABLE, | SEQNO NUMBER(5,0) NOT NULL ENABLE, |
| MESSAGE_CAT_NBR NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE, |
| MESSAGE_NBR NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE, |
| VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE, |
| CONSTRAINT F1_SYNC_REQ_IN_EXCP_FK FOREIGN KEY(F1_SYNC_REQ_IN_ID) REFERENCES F1_SYNC_REQ_IN ON DELETE CASCADE |

INDEX

CREATE UNIQUE INDEX F1T197P0 ON F1_SYNC_REQ_IN_EXCP(F1_SYNC_REQ_IN_ID,SEQNO) TABLESPACE CM_F1T191_IND

PARTITION BY REFERENCE (F1_SYNC_REQ_IN_EXCP_FK)
ENABLE ROW MOVEMENT;

ALTER TABLE F1_SYNC_REQ_IN_EXCP ADD CONSTRAINT F1T197P0 PRIMARY KEY (F1_SYNC_REQ_IN_ID,SEQNO) USING INDEX;

CREATE TABLE F1_SYNC_REQ_IN_EXCP_PARM

| F1_SYNC_REQ_IN_ID CHAR(14) NOT NULL ENABLE, | SEQNO NUMBER(5,0) NOT NULL ENABLE, |
| PARM_SEQ NUMBER(3,0) NOT NULL ENABLE, |
| MSG_PARM_VAL VARCHAR2(30) DEFAULT ' ' NOT NULL ENABLE, |
| MSG_PARM_TYP_FLG CHAR(4) DEFAULT ' ' NOT NULL ENABLE, |
| VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE, |
| CONSTRAINT F1_SYNC_REQ_IN_EXCP_PARM_FK FOREIGN KEY(F1_SYNC_REQ_IN_ID) REFERENCES F1_SYNC_REQ_IN ON DELETE CASCADE |

INDEX

CREATE UNIQUE INDEX F1T197P0 ON F1_SYNC_REQ_IN_EXCP_PARM(F1_SYNC_REQ_IN_ID,SEQNO) TABLESPACE CM_F1T191_IND

PARTITION BY REFERENCE (F1_SYNC_REQ_IN_EXCP_PARM_FK)
ENABLE ROW MOVEMENT;
### Child Table: F1_SYNC_REQ_IN_LOG

**CREATE TABLE F1_SYNC_REQ_IN_LOG**

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_SYNC_REQ_IN_ID</td>
<td>CHAR(14)</td>
<td></td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>SEQNO</td>
<td>NUMBER(5,0)</td>
<td></td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>LOG_ENTRY_TYPE_FLG</td>
<td>CHAR(4)</td>
<td>''</td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>LOG_DTTM</td>
<td>DATE</td>
<td></td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>BO_STATUS_CD</td>
<td>CHAR(12)</td>
<td>''</td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>MESSAGE_CAT_NBR</td>
<td>NUMBER(5,0)</td>
<td>0</td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>MESSAGE_NBR</td>
<td>NUMBER(5,0)</td>
<td>0</td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>CHAR_TYPE_CD</td>
<td>CHAR(8)</td>
<td>''</td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>CHAR_VAL</td>
<td>CHAR(16)</td>
<td>''</td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>ADHOC_CHAR_VAL</td>
<td>VARCHAR2(254)</td>
<td>''</td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>CHAR_VAL_FK1</td>
<td>VARCHAR2(50)</td>
<td>''</td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>CHAR_VAL_FK2</td>
<td>VARCHAR2(50)</td>
<td>''</td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>CHAR_VAL_FK3</td>
<td>VARCHAR2(50)</td>
<td>''</td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>CHAR_VAL_FK4</td>
<td>VARCHAR2(50)</td>
<td>''</td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>CHAR_VAL_FK5</td>
<td>VARCHAR2(50)</td>
<td>''</td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>DESCRLONG</td>
<td>VARCHAR2(4000)</td>
<td>''</td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>USER_ID</td>
<td>CHAR(8)</td>
<td>''</td>
<td>NOT NULL ENABLE</td>
</tr>
<tr>
<td>VERSION</td>
<td>NUMBER(5,0)</td>
<td>1</td>
<td>NOT NULL ENABLE</td>
</tr>
</tbody>
</table>

**CONSTRAINT F1_SYNC_REQ_IN_LOG_FK FOREIGN KEY** (F1_SYNC_REQ_IN_ID) REFERENCES F1_SYNC_REQ_IN ON DELETE CASCADE

**PARTITION BY REFERENCE** (F1_SYNC_REQ_IN_LOG_FK)

ENABLE ROW MOVEMENT;

---

**INDEX**

**CREATE UNIQUE INDEX F1T194P0 ON**

F1_SYNC_REQ_IN_LOG (F1_SYNC_REQ_IN_ID, SEQNO) TABLESPACE CM_F1T191_IND

**GLOBAL PARTITION BY RANGE (F1_SYNC_REQ_IN_ID)**

<table>
<thead>
<tr>
<th>Partition</th>
<th>Values Less Than</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>'124999999999'</td>
</tr>
<tr>
<td>P2</td>
<td>'249999999999'</td>
</tr>
<tr>
<td>P3</td>
<td>'374999999999'</td>
</tr>
<tr>
<td>P4</td>
<td>'499999999999'</td>
</tr>
<tr>
<td>P5</td>
<td>'624999999999'</td>
</tr>
<tr>
<td>P6</td>
<td>'749999999999'</td>
</tr>
</tbody>
</table>

**INDEX**

**CREATE UNIQUE INDEX F1T198P0 ON**

F1_SYNC_REQ_IN_EXCP_PARM (F1_SYNC_REQ_IN_ID, SEQNO, PARM_SEQ) TABLESPACE CM_F1T191_IND

**GLOBAL PARTITION BY RANGE (F1_SYNC_REQ_IN_ID)**

<table>
<thead>
<tr>
<th>Partition</th>
<th>Values Less Than</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>'124999999999'</td>
</tr>
<tr>
<td>P2</td>
<td>'249999999999'</td>
</tr>
<tr>
<td>P3</td>
<td>'374999999999'</td>
</tr>
<tr>
<td>P4</td>
<td>'499999999999'</td>
</tr>
<tr>
<td>P5</td>
<td>'624999999999'</td>
</tr>
<tr>
<td>P6</td>
<td>'749999999999'</td>
</tr>
<tr>
<td>P7</td>
<td>'874999999999'</td>
</tr>
<tr>
<td>P8</td>
<td>MAXVALUE</td>
</tr>
</tbody>
</table>

**COMPRESS ADVANCED LOW;**

**ALTER TABLE F1_SYNC_REQ_IN_EXCP_PARM ADD CONSTRAINT F1T198P0 PRIMARY KEY (F1_SYNC_REQ_IN_ID, SEQNO, PARM_SEQ) USING INDEX;**
Sample SQL for Enabling ILM in C2M for CC&B (Initial Install)

Oracle Utilities Customer To Meter
Database Administrator’s Guide

PARTITION P7 VALUES LESS THAN ( '874999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
) COMPRESS ADVANCED LOW;

ALTER TABLE F1_SYNC_REQ_IN_LOG ADD CONSTRAINT F1T194P0 PRIMARY KEY
(F1_SYNC_REQ_IN_ID,SEQNO) USING INDEX;

CREATE INDEX F1T194S1 ON F1_SYNC_REQ_IN_LOG(CHAR_TYPE_CD,CHAR_VAL_FK1)
TABLESPACE CM_F1T191_IND COMPRESS ADVANCED LOW;

CREATE INDEX F1T194S2 ON F1_SYNC_REQ_IN_LOG(CHAR_TYPE_CD,CHAR_VAL)
TABLESPACE CM_F1T191_IND COMPRESS ADVANCED LOW;

Child Table: F1_SYNC_REQ_IN_LOG_PARM

CREATE TABLE F1_SYNC_REQ_IN_LOG_PARM
(
  F1_SYNC_REQ_IN_ID CHAR(14) NOT NULL ENABLE,
  SEQNO             NUMBER(5,0) NOT NULL ENABLE,
  PARM_SEQ          NUMBER(3,0) NOT NULL ENABLE,
  MSG_PARM_VAL      VARCHAR2(30) DEFAULT ' ' NOT NULL ENABLE,
  MSG_PARM_TYP_FLG  CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
  VERSION           NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
  CONSTRAINT F1_SYNC_REQ_IN_LOG_PARM_FK FOREIGN KEY(F1_SYNC_REQ_IN_ID) REFERENCES F1_SYNC_REQ_IN ON DELETE CASCADE
) PARTITION BY REFERENCE (F1_SYNC_REQ_IN_ID) ENABLE ROW MOVEMENT;

INDEX
CREATE UNIQUE INDEX F1T195P0 ON
F1_SYNC_REQ_IN_LOG_PARM(F1_SYNC_REQ_IN_ID,SEQNO,PARM_SEQ) TABLESPACE
CM_F1T191_IND
GLOBAL PARTITION BY RANGE (F1_SYNC_REQ_IN_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 ( '749999999999' ),
  PARTITION P7 VALUES LESS THAN ( '874999999999' ),
  PARTITION P8 VALUES LESS THAN ( MAXVALUE )
) COMPRESS ADVANCED LOW;

ALTER TABLE F1_SYNC_REQ_IN_LOG_PARM ADD CONSTRAINT F1T195P0 PRIMARY
KEY (F1_SYNC_REQ_IN_ID,SEQNO,PARM_SEQ) USING INDEX;

Child Table: F1_SYNC_REQ_IN_REL_OBJ

CREATE TABLE F1_SYNC_REQ_IN_REL_OBJ
(
  F1_SYNC_REQ_IN_ID CHAR(14) NOT NULL ENABLE,
  MAINT_OBJ_CD      CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
  REL_OBJ_TYPE_FLG  CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
  PK_VALUE1         VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
PK_VALUE2         VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
PK_VALUE3         VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
PK_VALUE4         VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
PK_VALUE5         VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
VERSION           NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
CONSTRAINT F1_SYNC_REQ_IN_REL_OBJ_FK FOREIGN KEY(F1_SYNC_REQ_IN_ID)
REFERENCES F1_SYNC_REQ_IN ON DELETE CASCADE
)
PARTITION BY REFERENCE (F1_SYNC_REQ_IN_REL_OBJ_FK)
ENABLE ROW MOVEMENT;

INDEX
CREATE UNIQUE INDEX F1T192P0 ON
F1_SYNC_REQ_IN_REL_OBJ(F1_SYNC_REQ_IN_ID, MAINT_OBJ_CD,
REL_OBJ_TYPE_FLG) TABLESPACE CM_F1T191_IND
GLOBAL PARTITION BY RANGE (F1_SYNC_REQ_IN_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 ( '749999999999' ),
PARTITION P7 VALUES LESS THAN ( '874999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
)
COMPRESS ADVANCED LOW;

ALTER TABLE F1_SYNC_REQ_IN_REL_OBJ ADD CONSTRAINT F1T192P0 PRIMARY KEY
(F1_SYNC_REQ_IN_ID, MAINT_OBJ_CD, REL_OBJ_TYPE_FLG) USING INDEX;

CREATE INDEX F1T192S1 ON F1_SYNC_REQ_IN_REL_OBJ(PK_VALUE1) TABLESPACE
CM_F1T191_IND;
Appendix B

Sample SQL For Enabling ILM in C2M for CC&B (Existing Installation)

This section provides additional details related to supporting ILM in an existing installation. It includes 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:

   ```sql
   ALTER TABLE CI_TD_ENTRY RENAME TO ILM_TD_ENTRY;
   ALTER INDEX XT039P0 RENAME TO ILM_XT039P0;
   ```

2. Generate DDL for the secondary index.

   ```sql
   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.

   ```sql
   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.

   ```sql
   CREATE TABLE CI_TD_ENTRY ( TD_ENTRY_ID CHAR(14) NOT NULL ENABLE, BATCH_CD CHAR(8) DEFAULT ' ' NOT NULL ENABLE, BATCH_NBR NUMBER(10,0) DEFAULT 0 NOT NULL ENABLE, MESSAGE_CAT_NBR NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE, MESSAGE_NBR NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE, );
   ```
ASSIGNED_TO CHAR(8) DEFAULT '' NOT NULL ENABLE,
TD_TYPE_CD CHAR(8) DEFAULT '' NOT NULL ENABLE,
ROLE_ID CHAR(10) DEFAULT '' NOT NULL ENABLE,
ENTRY_STATUS_FLG CHAR(2) DEFAULT '' NOT NULL ENABLE,
VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
CRE_DTTM DATE,
ASSIGNED_DTTM DATE,
COMPLETE_DTTM DATE,
COMPLETE_USER_ID CHAR(8) DEFAULT '' NOT NULL ENABLE,
COMMENTS VARCHAR2(254) DEFAULT '' NOT NULL ENABLE,
ASSIGNED_USER_ID CHAR(8) DEFAULT '' NOT NULL ENABLE,
TD_PRIORITY_FLG CHAR(4) DEFAULT '' NOT NULL ENABLE,
ILM_DT DATE,
ILM_ARCH_SW CHAR(1)
)
DISABLE ROW MOVEMENT
PARTITION BY RANGE (ILM_DT)
SUBPARTITION BY RANGE (TD_ENTRY_ID) SUBPARTITION TEMPLATE
|
| SUBPARTITION S01 VALUES LESS THAN ('12499999999999999')
| SUBPARTITION S02 VALUES LESS THAN ('24999999999999999')
| SUBPARTITION S03 VALUES LESS THAN ('37499999999999999')
| SUBPARTITION S04 VALUES LESS THAN ('49999999999999999')
| SUBPARTITION S05 VALUES LESS THAN ('62499999999999999')
| SUBPARTITION S06 VALUES LESS THAN ('74999999999999999')
| SUBPARTITION S07 VALUES LESS THAN ('87499999999999999')
| SUBPARTITION SMAX VALUES LESS THAN (MAXVALUE)

PARTITION "P2017JAN" VALUES LESS THAN (TO_DATE('2017-02-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017JAN,
PARTITION "P2017FEB" VALUES LESS THAN (TO_DATE('2017-03-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017FEB,
PARTITION "P2017MAR" VALUES LESS THAN (TO_DATE('2017-04-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017MAR,
PARTITION "P2017APR" VALUES LESS THAN (TO_DATE('2017-05-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017APR,
PARTITION "P2017MAY" VALUES LESS THAN (TO_DATE('2017-06-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017MAY,
PARTITION "P2017JUN" VALUES LESS THAN (TO_DATE('2017-07-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017JUN,
PARTITION "P2017JUL" VALUES LESS THAN (TO_DATE('2017-08-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017JUL,
PARTITION "P2017AUG" VALUES LESS THAN (TO_DATE('2017-09-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017AUG,
PARTITION "P2017SEP" VALUES LESS THAN (TO_DATE('2017-10-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017SEP,
PARTITION "P2017OCT" VALUES LESS THAN (TO_DATE('2017-11-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017OCT,
PARTITION "P2017NOV" VALUES LESS THAN (TO_DATE('2017-12-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017NOV,
PARTITION "P2017DEC" VALUES LESS THAN (TO_DATE('2018-01-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017DEC,
PARTITION "PMAX" VALUES LESS THAN (MAXVALUE)
tablespace CM_XT039_PMAX
) as select /* PARALLEL */
as select /* PARALLEL */
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,
5. Enable logging option for table `CI_TD_ENTRY`.

   ```sql
   ALTER TABLE CI_TD_ENTRY NOPARALLEL LOGGING;
   ```

6. Create Primary Index for Parent table `CI_TD_ENTRY`.

   ```sql
   CREATE BIGFILE TABLESPACE CM_XT039_IND DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
   CREATE UNIQUE INDEX XT039P0 ON CI_TD_ENTRY NOLOGGING PARALLEL (TD_ENTRY_ID)
   PARTITION P1 VALUES LESS THAN ( '12499999999999' ),
   PARTITION P2 VALUES LESS THAN ( '24999999999999' ),
   PARTITION P3 VALUES LESS THAN ( '37499999999999' ),
   PARTITION P4 VALUES LESS THAN ( '49999999999999' ),
   PARTITION P5 VALUES LESS THAN ( '62499999999999' ),
   PARTITION P6 VALUES LESS THAN ( '74999999999999' ),
   PARTITION P7 VALUES LESS THAN ( '87499999999999' ),
   PARTITION P8 VALUES LESS THAN ( MAXVALUE ) TABLESPACE CM_XT039_IND;
   ALTER INDEX XT039P0 LOGGING NOPARALLEL;
   ```

7. Add Primary Key for Parent table `CI_TD_ENTRY`

   ```sql
   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`

   ```sql
   CREATE UNIQUE INDEX CM_ILM_XT039S8 ON CI_TD_ENTRY (IML_DT, ILM_ARCH_SW, TD_ENTRY_ID)
   LOCAL COMPRESS ADVANCED LOW /
   CREATE UNIQUE INDEX XT039S2 ON CI_TD_ENTRY (ASSIGNED_TO, TD_ENTRY_ID) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW /
   CREATE UNIQUE INDEX XT039S3 ON CI_TD_ENTRY (ENTRY_STATUS_FLG, ASSIGNED_TO) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW /
   CREATE INDEX XT039S4 ON CI_TD_ENTRY (ROLE_ID, TD_TYPE_CD, ENTRY_STATUS_FLG, TD_PRIORITY_FLG) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW /
   CREATE INDEX XT039S5 ON CI_TD_ENTRY (BATCH_CD, BATCH_NBR, ENTRY_STATUS_FLG) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW /
   CREATE UNIQUE INDEX XT039S6 ON CI_TD_ENTRY (TD_ENTRY_ID, ASSIGNED_TO, ENTRY_STATUS_FLG) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW /
   CREATE UNIQUE INDEX XT039S7 ON CI_TD_ENTRY (COMPLETE_USER_ID, COMPLETE_DTTM, TD_ENTRY_ID) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW /
   ```

9. After verification of the ILM based tables, user can drop the backup tables “ILM” renamed table.

10. Create all child Tables, Primary Key, Primary Indexes and Secondary Indexes as shown below.

   Repeat the following steps for all child tables.

   **Create Child Table CI_TD_DRLKEY**

   ```sql
   CREATE TABLE CI_TD_DRLKEY
   ( TD_ENTRY_ID NOT NULL ENABLE,
   SEQ_NUM    NOT NULL ENABLE,
   /```
KEY_VALUE DEFAULT ' ' NOT NULL ENABLE,
VERSION DEFAULT 1 NOT NULL ENABLE,
CONSTRAINT CI_TD_DRLKEY_FK FOREIGN KEY(TD_ENTRY_ID) REFERENCES CI_TD_ENTRY ON DELETE CASCADE
)
PARTITION BY REFERENCE (CI_TD_DRLKEY_FK)
ENABLE ROW MOVEMENT
AS SELECT /*+ PARALLEL */ * FROM ILM_CI_TD_DRLKEY;

Create Index

CREATE UNIQUE INDEX XT037P0 ON CI_TD_DRLKEY ( TD_ENTRY_ID, SEQ_NUM ) TABLESPACE CM_XT039_IND NOLOGGING PARALLEL
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 ( '749999999999' ),
PARTITION P7 VALUES LESS THAN ( '874999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
}
COMPRESS ADVANCED LOW;
ALTER INDEX XT037P0 LOGGING NOPARALLEL;
ALTER TABLE CI_TD_DRLKEY ADD CONSTRAINT XT037P0 PRIMARY KEY(TD_ENTRY_ID, SEQ_NUM) USING INDEX;
CREATE INDEX XT037S1 ON CI_TD_DRLKEY ( KEY_VALUE, TD_ENTRY_ID ) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;
Appendix C
Sample SQL for Periodic Maintenance for CC&B Data

This appendix provides additional details related to creating new partitions over time as well as archiving and restoring partitions. The To Do Entry and Inbound Sync Request maintenance objects are used as examples. This section contains the following steps:

- Add Partition
- Archive Partition
- Restore Partition
Add Partition

1. Create separate tablespace for new partition
   
   ```sql
   CREATE BIGFILE TABLESPACE CM_XT039_P2016JAN DATAFILE '+DATA' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
   ```
   
2. Add partition using split operation on MAXVALUE Partition
   
   ```sql
   ALTER TABLE CISADM.CI_TD_ENTRY SPLIT PARTITION PMAX AT (TO_DATE('2016-02-01 00:00:01','SYYYY-MM-DD HH24:MI:SS')) INTO
   (PARTITION P2016JAN TABLESPACE CM_XT039_P2016JAN, PARTITION PMAX )
   UPDATE INDEXES;
   ```
   
   In case table contains LOBS like F1_SYNC_REQ_IN, there will be additional statement in split partition DDL indicating tablespace on which LOB should go.
   
   ```sql
   ALTER TABLE CISADM.F1_SYNC_REQ_IN SPLIT PARTITION PMAX AT (TO_DATE('2016-02-01 00:00:01','SYYYY-MM-DD HH24:MI:SS')) INTO
   (PARTITION P2016JAN TABLESPACE CM_F1T191_P2016JAN
   LOB(BO_DATA_AREA, POST_TRN_BO_DATA_AREA, PRE_TRN_FIN_BO_DATA_AREA, PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2016JAN )
   , PARTITION PMAX)
   UPDATE INDEXES;
   ```
   
3. Enable advanced compression after SPLIT partition as it will disable the compression.
   
   ```sql
   ALTER TABLE CISADM.CI_TD_SRTKEY ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_MSG_PARM ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_DRLKEY ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_ENTRY_CHA ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_LOG ROW STORE COMPRESS ADVANCED;
   ```

Archive Partition

1. Make the tablespace to be archived READ ONLY.
   
   ```sql
   ALTER TABLESPACE CM_XT039_P2017JAN READ ONLY;
   ```
   
2. Check the feasibility of archive using ILM_ARCH_SW = ‘N’.
   
   ```sql
   Select count(1) from CISADM.CI_TD_ENTRY PARTITION P2017JAN where ILM_ARCH_SW = ‘N’;
   ```
   
   • If Yes (count of records of above query is ZERO), then proceed for further steps.
   • If No (count of records of above query is Non ZERO), then make the tablespace back to READ WRITE MODE as Archive is not Feasible at the time.
ALTER TABLESPACE CM_XT039_P2017JAN READ WRITE;

3. Create separate archive tablespace for partition need to be archived.

CREATE BIGFILE TABLESPACE CM_XT039_P2017JAN_ARC DATAFILE '+DATA' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;

4. Create staging tables and load data for all child tables for the MO first.

   a. CI_TD_ENTRY_CHA

      CREATE TABLE CM_XT701_P2017JAN_ARC PARALLEL NOLOGGING TABLESPACE CM_XT039_P2017JAN_ARC AS
      (
         SELECT /*+ PARALLEL */ * FROM CISADM.CI_TD_ENTRY_CHA PARTITION (P2017JAN_S01)
         UNION ALL
         SELECT /*+ PARALLEL */ * FROM CI_TD_ENTRY_CHA PARTITION (P2017JAN_S02)
         UNION ALL...
         UNION ALL
         SELECT /*+ PARALLEL */ * FROM CI_TD_ENTRY_CHA PARTITION (P2017JAN_S08)
      );
      ALTER TABLE CM_XT701_P2017JAN_ARC NOPARALLEL LOGGING;

   b. CI_TD_MSG_PARM

      CREATE TABLE CM_XT04_P2017JAN_ARC PARALLEL NOLOGGING TABLESPACE CM_XT039_P2017JAN_ARC AS
      (
         SELECT /*+ PARALLEL */ * FROM CISADM.CI_TD_MSG_PARM PARTITION (P2017JAN_S01)
         UNION ALL
         SELECT /*+ PARALLEL */ * FROM CI_TD_MSG_PARM PARTITION (P2017JAN_S02)
         UNION ALL...
         UNION ALL
         SELECT /*+ PARALLEL */ * FROM CI_TD_MSG_PARM PARTITION (P2017JAN_S08)
      );
      ALTER TABLE CM_XT04_P2017JAN_ARC NOPARALLEL LOGGING;

   c. CI_TD_LOG

      CREATE TABLE CM_XT721_P2017JAN_ARC PARALLEL NOLOGGING TABLESPACE CM_XT039_P2017JAN_ARC AS
      (
         SELECT /*+ PARALLEL */ * FROM CISADM.CI_TD_LOG PARTITION (P2017JAN_S01)
         UNION ALL...
Sample SQL for Periodic Maintenance for CC&B Data

   ```sql
   ALTER TABLESPACE CM_XT039_P2017JAN_ARC READ ONLY;
   expdp system/manager DIRECTORY=DUMP_DIR DUMPFILE=
   CM_XT039_P2017JAN_ARC.DMP TRANSPORT_TABLESPACES=
   CM_XT039_P2017JAN_ARC LOGFILE=EXP_CM_XT039_P2017JAN_ARC.LOG
   TRANSPORT_FULL_CHECK=Y
   ```

   Ensure tablespace datafile required for further import should be preserved.

   ```sql
   <<Transport THE FILE to LOCAL DB DIRECTORY DUMP_DIR like
   connected to asmcmd and copied the file from cp
   cm_xt039_p201701_tbs_ar.553.913864937 /tugbu_perf_02/BACKUPS/
   test_verification/ >>
   ```

7. Drop the partition, partition tablespace and archive tablespace (as it is already exported).

   ```sql
   ALTER TABLE CISADM.CI_TD_ENTRY DROP PARTITION P2017JAN UPDATE
   INDEXES;
   DROP TABLESPACE CM_XT039_P2017JAN INCLUDING CONTENTS AND
   DATAFILES;
   DROP TABLESPACE CM_XT039_P2017JAN_ARC INCLUDING CONTENTS AND
   DATAFILES;
   ```

 Restore Partition

1. Create separate tablespace to restore the partition.

   ```sql
   CREATE BIGFILE TABLESPACE CM_XT039_P2017JAN DATAFILE '+DATA'
   SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE
   COMPRESS ADVANCED;
   ```

2. Add partition using split operation on next greater value partition

   ```sql
   ALTER TABLE CISADM.CI_TD_ENTRY SPLIT PARTITION P2017FEB AT
   (TO_DATE('2017-02-01 00:00:01','SYYYY-MM-DD HH24:MI:SS'))
   INTO
   ( PARTITION P2017JAN TABLESPACE CM_XT039_P2017JAN , PARTITION
   P2017FEB
   )
   UPDATE INDEXES;
   ```

   In case table contains LOBS like F1_SYNC_REQ_IN, there will be additional statement in
   split partition DDL indicating tablespace on which LOB should go.

   ```sql
   ALTER TABLE CISADM.F1_SYNC_REQ_IN SPLIT PARTITION P2017FEB AT
   (TO_DATE('2017-02-01 00:00:01','SYYYY-MM-DD HH24:MI:SS'))
   INTO
   ( PARTITION P2017JAN TABLESPACE CM_F1T191_P2017JAN
     LOB(BO_DATA_AREA,PRE_TRN_INIT_BO_DATA_AREA,PRE_TRN_FIN_BO_DATA_       
     AREA,POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE
     IN ROW COMPRESS MEDIUM CACHE TABLESPACE
     CM_F1T191_P2017JAN )
   , PARTITION P2017FEB
   ```
3. Enable advanced compression after SPLIT partition as it will disable the compression.

   ALTER TABLE CISADM.CI_TD_SRTKEY ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_MSG_PARM ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_DRLKEY ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_ENTRY_CHA ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_LOG ROW STORE COMPRESS ADVANCED;

4. Import tablespace using TRANSPORT_TABLESPACES method.

   impdp system/manager DIRECTORY=DUMP_DIR DUMPFILE=
   CM_XT039_P2017JAN_ARC.DMP PARTITION_OPTIONS=DEPARTITION
   LOGFILE=IMP_CM_XT039_P2017JAN_ARC.LOG TRANSPORT_DATAFILES=/
   tugbu_perf_02/BACKUPS/test_verification/
   cm_xt039_p201701jan_ar.553.913864937

5. Load data into parent table first from the staging table

   INSERT /*+ APPEND PARALLEL */ INTO CISADM.CI_TD_ENTRY
   SELECT /*+ PARALLEL */ * FROM CM_XT039_P2017JAN_ARC;
   COMMIT;

6. Load data into child table from the staging table

   For each Child IN LIST OF CHILD TABLES, perform the following:

   INSERT /*+ APPEND PARALLEL */ INTO CISADM.CI_TD_ENTRY_CHA
   SELECT /*+ PARALLEL */ * FROM CM_XT701_P2017JAN_ARC;
   COMMIT;

   INSERT /*+ APPEND PARALLEL */ INTO CISADM.CI_TD_MSG_PARM
   SELECT /*+ PARALLEL */ * FROM CM_XT04_P2017JAN_ARC;
   COMMIT;

   INSERT /*+ APPEND PARALLEL */ INTO CISADM.CI_TD_LOG
   SELECT /*+ PARALLEL */ * FROM CM_XT721_P2017JAN_ARC;
   COMMIT;

   INSERT /*+ APPEND PARALLEL */ INTO CISADM.CI_TD_SRTKEY
   SELECT /*+ PARALLEL */ * FROM CM_XT041_P2017JAN_ARC;
   COMMIT;

   INSERT /*+ APPEND PARALLEL */ INTO CISADM.CI_TD_DRLKEY
   SELECT /*+ PARALLEL */ * FROM CM_XT037_P2017JAN_ARC;
   COMMIT;

7. Drop the archive tablespace after import is import and data loading is successful.

   DROP TABLESPACE CM_XT039_P2017JAN_ARC INCLUDING CONTENTS AND
   DATAFILES;
These are the sample SQL scripts which has the recommended way to partition the Bill Segment and Adjustment tables and Indexes. Implementations can further customize these scripts and update the partition names, tablespace names and date ranges to make sure they are suited to the implementation.

This appendix consists:

- Maintenance Object: Adjustment
- Maintenance Object: Bill Segment
This section contains the sample SQL for the following tables:

- **Parent Table: CI_ADJ**
  - Child Table: CI_ADJ_APREQ
  - Child Table: CI_ADJ_CALC_LN
    - Child Table: CI_ADJ_CL_CHAR
  - Child Table: CI_ADJ_CHAR

**Parent Table: CI_ADJ**

```sql
CREATE BIGFILE TABLESPACE CM_XT012_P2017JAN DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT012_P2017FEB DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT012_P2017MAR DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT012_P2017APR DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT012_P2017MAY DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT012_P2017JUN DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT012_P2017JUL DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT012_P2017AUG DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT012_P2017SEP DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT012_P2017OCT DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT012_P2017NOV DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT012_P2017DEC DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE TABLE CI_ADJ
  ( CHAR(12) NOT NULL ENABLE,
    ID CHAR(10) DEFAULT ' ' NOT NULL ENABLE,
    ADJ_TYPE_CCHAR(8) DEFAULT ' ' NOT NULL ENABLE,
    ADJ_STATUS_FLG CHAR(2) DEFAULT ' ' NOT NULL ENABLE,
```

---

Sample SQL for Partitioning with ILM in C2M for CC&B D - 2
Oracle Utilities Customer To Meter Database Administrator’s Guide
Sample SQL for Partitioning with ILM in C2M for CC&B D - 3

Oracle Utilities Customer To Meter Database Administrator's Guide

CRE_DT DATE, CHAR(4) DEFAULT '' NOT NULL ENABLE,
CAN_RSN_CD
ADJ_AMT NUMBER(15,2) DEFAULT 0 NOT NULL ENABLE,
CURRENCY_CD CHAR(3) DEFAULT '' NOT NULL ENABLE,
COMMENTS VARCHAR2(254) DEFAULT '' NOT NULL ENABLE,
VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
BEHALF_SA_ID CHAR(10) DEFAULT '' NOT NULL ENABLE,
BASE_AMT NUMBER(15,2) DEFAULT 0 NOT NULL ENABLE,
GEN_REF_DT DATE,
APPR_REQ_ID CHAR(12) DEFAULT '' NOT NULL ENABLE,
ADJ_DATA_AREA CLOB,
ILM_DT DATE,
ILM_ARCH_SW CHAR(1),
)

ENABLE ROW MOVEMENT

PARTITION BY RANGE (ILM_DT)
SUBPARTITION BY RANGE (ADJ_ID) SUBPARTITION TEMPLATE ( SUBPARTITION S01 VALUES LESS THAN ('124999999999'),
SUBPARTITION S02 VALUES LESS THAN ('249999999999'),
SUBPARTITION S03 VALUES LESS THAN ('374999999999'),
SUBPARTITION S04 VALUES LESS THAN ('499999999999'),
SUBPARTITION S05 VALUES LESS THAN ('624999999999'),
SUBPARTITION S06 VALUES LESS THAN ('749999999999'),
SUBPARTITION S07 VALUES LESS THAN ('874999999999'),
SUBPARTITION S08 VALUES LESS THAN (MAXVALUE)
)

{
PARTITION "P2017JAN" VALUES LESS THAN (TO_DATE('2017-02-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017JAN,
PARTITION "P2017FEB" VALUES LESS THAN (TO_DATE('2017-03-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017FEB,
PARTITION "P2017MAR" VALUES LESS THAN (TO_DATE('2017-04-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017MAR,
PARTITION "P2017APR" VALUES LESS THAN (TO_DATE('2017-05-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017APR,
PARTITION "P2017MAY" VALUES LESS THAN (TO_DATE('2017-06-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017MAY,
PARTITION "P2017JUN" VALUES LESS THAN (TO_DATE('2017-07-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017JUN,
PARTITION "P2017JUL" VALUES LESS THAN (TO_DATE('2017-08-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017JUL,
PARTITION "P2017AUG" VALUES LESS THAN (TO_DATE('2017-09-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017AUG,
PARTITION "P2017SEP" VALUES LESS THAN (TO_DATE('2017-10-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017SEP,
PARTITION "P2017OCT" VALUES LESS THAN (TO_DATE('2017-11-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017OCT,
PARTITION "P2017NOV" VALUES LESS THAN (TO_DATE('2017-12-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017NOV,
PARTITION "P2017DEC" VALUES LESS THAN (TO_DATE('2018-01-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT012_P2017DEC,
PARTITION "PMAX" VALUES LESS THAN (MAXVALUE)
tablespace CM_XT012_PMAX
)
/

INDEX

CREATE BIGFILE TABLESPACE CM_XT012_IND DATAFILE '+DATA' SIZE 50M
AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED
/
CREATE UNIQUE INDEX XT012P0 ON CI_ADJ ( ADJ_ID ) TABLESPACE
CM_XT012_IND
GLOBAL PARTITION BY RANGE (ADJ_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 ( '749999999999' ),
PARTITION P7 VALUES LESS THAN ( '874999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
}
/

ALTER TABLE CI_ADJ ADD CONSTRAINT XT012P0 PRIMARY KEY(ADJ_ID) USING
INDEX
/
CREATE INDEX XT012S1 ON CI_ADJ ( SA_ID, ADJ_TYPE_CD ) TABLESPACE
CM_XT012_IND COMPRESS ADVANCED LOW
/
CREATE UNIQUE INDEX XT012S2 ON CI_ADJ ( XFER_ADJ_ID, ADJ_ID )
TABLESPACE CM_XT012_IND COMPRESS ADVANCED LOW
/
CREATE UNIQUE INDEX XT012S3 ON CI_ADJ ( ILM_DT, ILM_ARCH_SW, ADJ_ID )
TABLESPACE CM_XT012_IND COMPRESS ADVANCED LOW
/

Child Table: CI_ADJ_APREQ

CREATE TABLE CI_ADJ_APREQ
{
AP_REQ_ID      CHAR(12) NOT NULL ENABLE,
COUNTRY        CHAR(3) DEFAULT ' ' NOT NULL ENABLE,
ADDRESS1       VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
ADJ_ID         CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
ADDRESS2       VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
ADDRESS3       VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
ADDRESS4       VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
CITY           VARCHAR2(90) DEFAULT ' ' NOT NULL ENABLE,
NUM1           CHAR(6) DEFAULT ' ' NOT NULL ENABLE,
}
### Parent Table: CI_ADJ

```sql
NUM2 CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
COUNTY VARCHAR2(90) DEFAULT ' ' NOT NULL ENABLE,
HOUSE_TYPE CHAR(2) DEFAULT ' ' NOT NULL ENABLE,
STATE CHAR(6) DEFAULT ' ' NOT NULL ENABLE,
POSTAL CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
CURRENCY_PYMNT CHAR(3) DEFAULT ' ' NOT NULL ENABLE,
GEO_CODE CHAR(11) DEFAULT ' ' NOT NULL ENABLE,
IN_CITY_LIMIT CHAR(1) DEFAULT ' ' NOT NULL ENABLE,
PAID_AMT NUMBER(15,2) DEFAULT 0 NOT NULL ENABLE,
SCHEDULED_PAY_DT DATE,
PYMNT_DT DATE,
ENTITY_NAME VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
PAY_DOC_ID VARCHAR2(20) DEFAULT ' ' NOT NULL ENABLE,
PAY_DOC_DT DATE,
PYMNT_ID CHAR(36) DEFAULT ' ' NOT NULL ENABLE,
PYMNT_METHOD_FLG CHAR(3) DEFAULT ' ' NOT NULL ENABLE,
PYMNT_SEL_STAT_FLG CHAR(1) DEFAULT ' ' NOT NULL ENABLE,
VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
BATCH_CD CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
BATCH_NBR NUMBER(10,0) DEFAULT 0 NOT NULL ENABLE,
CONSTRAINT CI_ADJ_APREQ_FK FOREIGN KEY(ADJ_ID) REFERENCES CI_ADJ ON DELETE CASCADE
)
PARTITION BY REFERENCE (CI_ADJ_APREQ_FK) REFERENCES CI_ADJ
ENABLE ROW MOVEMENT
/

INDEX
CREATE UNIQUE INDEX XT160P0 ON CI_ADJ_APREQ (AP_REQ_ID) TABLESPACE CM_XT012_IND
GLOBAL PARTITION BY RANGE (AP_REQ_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 ('749999999999'),
PARTITION P7 VALUES LESS THAN ('874999999999'),
PARTITION P8 VALUES LESS THAN (MAXVALUE)
}
COMPRESS ADVANCED LOW
/
ALTER TABLE CI_ADJ_APREQ ADD CONSTRAINT XT160P0 PRIMARY KEY(AP_REQ_ID) USING INDEX
/
CREATE INDEX XT160S1 ON CI_ADJ_APREQ (ADJ_ID) TABLESPACE CM_XT012_IND
/
CREATE INDEX XT160S2 ON CI_ADJ_APREQ (BATCH_CD, BATCH_NBR) TABLESPACE CM_XT012_IND COMPRESS ADVANCED LOW
/

### Child Table: CI_ADJ_CALC_LN

CREATE TABLE CI_ADJ_CALC_LN
(
ADJ_ID CHAR(12) NOT NULL ENABLE,
SEQNO NUMBER(5,0) NOT NULL ENABLE,
)
CREATE TABLE CI_ADJ_CL_CHAR
(
  ADJ_ID         CHAR(12) NOT NULL ENABLE,
  SEQNO          NUMBER(5,0) NOT NULL ENABLE,
  CHAR_TYPE_CD   CHAR(8) NOT NULL ENABLE,
  VERSION        NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
  CHAR_VAL       CHAR(16) DEFAULT ' ' NOT NULL ENABLE,
  ADHOC_CHAR_VAL VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
  CONSTRAINT CI_ADJ_CL_CHAR_PK PRIMARY KEY(ADJ_ID)
);
Parent Table: CI_ADJ

```
CHAR_VAL_FK1   VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK2   VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK3   VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK4   VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK5   VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CONSTRAINT CI_ADJ_CL_CHAR_FK FOREIGN KEY(ADJ_ID) REFERENCES CI_ADJ ON
DELETE CASCADE
)
PARTITION BY REFERENCE (CI_ADJ_CL_CHAR_FK)
ENABLE ROW MOVEMENT
/

INDEX

CREATE UNIQUE INDEX XT309P0 ON CI_ADJ_CL_CHAR ( ADJ_ID, SEQNO,
CHAR_TYPE_CD ) TABLESPACE CM_XT012_IND
GLOBAL PARTITION BY RANGE (ADJ_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 ( '749999999999' ),
PARTITION P7 VALUES LESS THAN ( '874999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
)
COMPRESS ADVANCED LOW
/
ALTER TABLE CI_ADJ_CL_CHAR ADD CONSTRAINT XT309P0 PRIMARY KEY(ADJ_ID,
SEQNO, CHAR_TYPE_CD) USING INDEX
/

Child Table: CI_ADJ_CHAR

CREATE TABLE CI_ADJ_CHAR
( ADJ_ID          CHAR(12) NOT NULL ENABLE,
CHAR_TYPE_CD     CHAR(8) NOT NULL ENABLE,
SEQ_NUM         NUMBER(3,0) NOT NULL ENABLE,
VERSION         NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
CHAR_VAL        CHAR(16) DEFAULT ' ' NOT NULL ENABLE,
ADHOC_CHAR_VAL  VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK1    VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK2    VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK3    VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK4    VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK5    VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
SRCH_CHAR_VAL   VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CONSTRAINT CI_ADJ_CHAR_FK FOREIGN KEY(ADJ_ID) REFERENCES CI_ADJ ON
DELETE CASCADE
)
PARTITION BY REFERENCE (CI_ADJ_CHAR_FK)
ENABLE ROW MOVEMENT
/

INDEX
```
CREATE UNIQUE INDEX XC781P0 ON CI_ADJ_CHAR (ADJ_ID, CHAR_TYPE_CD, SEQ_NUM) TABLESPACE CM_XT012_IND
GLOBAL PARTITION BY RANGE(ADJ_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)
)
COMPRESS ADVANCED LOW
/

ALTER TABLE CI_ADJ_CHAR ADD CONSTRAINT XC781P0 PRIMARY KEY (ADJ_ID, CHAR_TYPE_CD, SEQ_NUM) USING INDEX;

CREATE INDEX XC781S1 ON CI_ADJ_CHAR(SRCH_CHAR_VAL) GLOBAL PARTITION BY HASH(SRCH_CHAR_VAL)
(
    PARTITION PART1 TABLESPACE CM_XT012_IND,
    PARTITION PART2 TABLESPACE CM_XT012_IND,
    PARTITION PART3 TABLESPACE CM_XT012_IND,
    PARTITION PART4 TABLESPACE CM_XT012_IND,
    PARTITION PART5 TABLESPACE CM_XT012_IND,
    PARTITION PART6 TABLESPACE CM_XT012_IND,
    PARTITION PART7 TABLESPACE CM_XT012_IND,
    PARTITION PART8 TABLESPACE CM_XT012_IND
)
/
Maintenance Object: Bill Segment

This section contains the sample SQL for the following tables:

- Parent Table: CI_BSEG
  - Child Table: CI_BSEG_CALC
  - Child Table: CI_BSEG_CALC_LN
    - Child Table: CI_BSEG_CL_CHAR
    - Child Table: CI_BSEG_EXCP
  - Child Table: CI_BSEG_MSG
  - Child Table: CI_BSEG_READ
  - Child Table: CI_BSEG_SQ
  - Child Table: CI_BSEG_ITEM

Parent Table: CI_BSEG

```sql
CREATE BIGFILE TABLESPACE CM_XT048_P2017JAN DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT048_P2017FEB DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT048_P2017MAR DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT048_P2017APR DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT048_P2017MAY DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT048_P2017JUN DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT048_P2017JUL DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT048_P2017AUG DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT048_P2017SEP DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT048_P2017OCT DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT048_P2017NOV DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT048_P2017DEC DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
CREATE BIGFILE TABLESPACE CM_XT048_PMAX DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED /
```
CREATE TABLE CI_BSEG
( CHAR(12) NOTNULL ENABLE,
BSEG_ID
BILL_CYC_CD CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
WIN_START_DT DATE,
CAN_RSN_CD CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
CAN_BSEG_ID CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
SA_ID CHAR(10) DEFAULT ' ' NOT NULL ENABLE,
BILL_ID CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
START_DT DATE,
END_DT DATE, CHAR(1) DEFAULT ' ' NOT NULL ENABLE,
EST_SW
CLOSING_BSEG_SW CHAR(1) DEFAULT ' ' NOT NULL ENABLE,
SQ_OVERRIDE_SW CHAR(1) DEFAULT ' ' NOT NULL ENABLE,
ITEM_OVERRIDE_SW CHAR(1) DEFAULT ' ' NOT NULL ENABLE,
PREM_ID CHAR(10) DEFAULT ' ' NOT NULL ENABLE,
BSEG_STAT_FLG CHAR(2) DEFAULT ' ' NOT NULL ENABLE,
CRE_DTTM DATE,
STAT_CHG_DTTM DATE,
REBILL_SEG_ID CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
VERSIONNUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
MASTER_BSEG_ID CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
QUOTE_DTL_ID CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
BILL_SCNR_ID CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
MDM_START_DTTM DATE,
MDM_END_DTTM DATE,
BSEG_DATA_AREA CLOB,
ILM_DT DATE,
ILM_ARCH_SW CHAR(1)
) ENABLE ROW MOVEMENT
PARTITION BY RANGE (ILM_DT)
SUBPARTITION BY RANGE (BSEG_ID) SUBPARTITION TEMPLATE ( 
SUBPARTITION S01 VALUES LESS THAN ( '124999999999' ), 
SUBPARTITION S02 VALUES LESS THAN ( '249999999999' ), 
SUBPARTITION S03 VALUES LESS THAN ( '374999999999' ), 
SUBPARTITION S04 VALUES LESS THAN ( '499999999999' ), 
SUBPARTITION S05 VALUES LESS THAN ( '624999999999' ), 
SUBPARTITION S06 VALUES LESS THAN ( '749999999999' ), 
SUBPARTITION S07 VALUES LESS THAN ( '874999999999' ), 
SUBPARTITION S08 VALUES LESS THAN ( MAXVALUE ) 
)

( PARTITION "P2017JAN" VALUES LESS THAN (TO_DATE('2017-02-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=Gregorian'))
tablespace CM_XT048_P2017JAN,
PARTITION "P2017FEB" VALUES LESS THAN (TO_DATE('2017-03-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=Gregorian'))
tablespace CM_XT048_P2017FEB,
PARTITION "P2017MAR" VALUES LESS THAN (TO_DATE('2017-04-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=Gregorian'))
tablespace CM_XT048_P2017MAR,
PARTITION "P2017APR" VALUES LESS THAN (TO_DATE('2017-05-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=Gregorian'))
tablespace CM_XT048_P2017APR,
PARTITION "P2017MAY" VALUES LESS THAN (TO_DATE('2017-06-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=Gregorian'))
tablespace CM_XT048_P2017MAY,
PARTITION "P2017JUN" VALUES LESS THAN (TO_DATE('2017-07-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=Gregorian'))
)

CREATE BIGFILE TABLESPACE CM_XT048_IND DATAFILE '+DATA' SIZE 100M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;

INDEX
CREATE UNIQUE INDEX XT048P0 ON CI_BSEG ( BSEG_ID ) TABLESPACE CM_XT048_IND
GLOBAL PARTITION BY RANGE (BSEG_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 ( '749999999999' ),
PARTITION P7 VALUES LESS THAN ( '874999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
}

ALTER TABLE CI_BSEG ADD CONSTRAINT XT048P0 PRIMARY KEY(BSEG_ID) USING INDEX

CREATE INDEX XT048S1 ON CI_BSEG ( BILL_ID ) TABLESPACE CM_XT048_IND

CREATE INDEX XT048S2 ON CI_BSEG ( SA_ID ) TABLESPACE CM_XT048_IND

CREATE UNIQUE INDEX XT048S3 ON CI_BSEG ( QUOTE_DTL_ID, BSEG_ID ) TABLESPACE CM_XT048_IND COMPRESS ADVANCED LOW

CREATE UNIQUE INDEX XT048S4 ON CI_BSEG ( ILM_DT, ILM_ARCH_SW, BSEG_ID ) TABLESPACE CM_XT048_IND COMPRESS ADVANCED LOW

Child Table: CI_BSEG_CALC
CREATE TABLE CI_BSEG_CALC
{
BSEG_ID CHAR(12) NOT NULL ENABLE,
CREATE TABLE CI_BSEG_CALC_LN
(
  BSEG_ID      CHAR(12) NOT NULL ENABLE,
  HEADER_SEQ   NUMBER(3,0) NOT NULL ENABLE,
  SEQNO        NUMBER(5,0) NOT NULL ENABLE,
  CHAR_TYPE_CD CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
  CURRENCY_CD  CHAR(3) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL     CHAR(16) DEFAULT ' ' NOT NULL ENABLE,
  DST_ID       CHAR(10) DEFAULT ' ' NOT NULL ENABLE,
  UOM_CD       CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
  TOU_CD       CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
  RC_SEQ       NUMBER(4,0) DEFAULT 0 NOT NULL ENABLE,
  PRT_SW       CHAR(1) DEFAULT ' ' NOT NULL ENABLE,
  APP_IN_SUMM_SW CHAR(1) DEFAULT ' ' NOT NULL ENABLE,
  CALC_AMT     NUMBER(15,2) DEFAULT 0 NOT NULL ENABLE,
  EXEMPT_AMT   NUMBER(15,2) DEFAULT 0 NOT NULL ENABLE,
  BASE_AMT     NUMBER(15,2) DEFAULT 0 NOT NULL ENABLE,
  SQI_CD       CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
)

Parent Table: CI_BSEG

```
BILLSQ NUMBER(18,6) DEFAULT 0 NOT NULL ENABLE,
MSR_PEAQ_QTY_SW CHAR(1) DEFAULT ' ' NOT NULL ENABLE,
DESCR_ON_BILL VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
AUDIT_CALC_AMT NUMBER(18,5) DEFAULT 0 NOT NULL ENABLE,
CALC_GRP_CD VARCHAR2(30) DEFAULT ' ' NOT NULL ENABLE,
CALC_RULE_CD VARCHAR2(30) DEFAULT ' ' NOT NULL ENABLE,
CONSTRAINT CI_BSEG_CALC_LN_FK FOREIGN KEY(BSEG_ID) REFERENCES CI_BSEG
ON DELETE CASCADE
)
PARTITION BY REFERENCE (CI_BSEG_CALC_LN_FK)
ENABLE ROW MOVEMENT
/

INDEX
CREATE UNIQUE INDEX XT050P0 ON CM_BSEG_CALC_LN (BSEG_ID,
HEADER_SEQ,SEQNO) TABLESPACE CM_XT048_IND
GLOBAL PARTITION BY RANGE (BSEG_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 ( '749999999999' ),
PARTITION P7 VALUES LESS THAN ( '874999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
}
COMPRRESS ADVANCED LOW
/

ALTER TABLE CI_BSEG_CALC_LN ADD CONSTRAINT XT050P0 PRIMARY
KEY(BSEG_ID, HEADER_SEQ,SEQNO) USING INDEX
/

Child Table: CI_BSEG_CL_CHAR

CREATE TABLE CI_BSEG_CL_CHAR
(
  BSEG_ID CHAR(12) NOT NULL ENABLE,
  HEADER_SEQ NUMBER(3,0) NOT NULL ENABLE,
  SEQNO NUMBER(5,0) NOT NULL ENABLE,
  CHAR_TYPE_CD CHAR(8) NOT NULL ENABLE,
  CHAR_VAL CHAR(16) DEFAULT ' ' NOT NULL ENABLE,
  ADHOC_CHAR_VAL VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK1 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK2 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK3 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK4 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK5 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
CONSTRAINT CI_BSEG_CL_CHAR_FK FOREIGN KEY(BSEG_ID) REFERENCES CI_BSEG
ON DELETE CASCADE
)
PARTITION BY REFERENCE (CI_BSEG_CL_CHAR_FK)
ENABLE ROW MOVEMENT
/
INDEX
CREATE UNIQUE INDEX XT056P0 ON CI_BSEG_CL_CHAR ( BSEG_ID, HEADER_SEQ, SEQNO, CHAR_TYPE_CD ) TABLESPACE CM_XT048_IND
GLOBAL PARTITION BY RANGE (BSEG_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 ( '749999999999' ),
  PARTITION P7 VALUES LESS THAN ( '874999999999' ),
  PARTITION P8 VALUES LESS THAN ( MAXVALUE )
} COMPRESS ADVANCED LOW
/

ALTER TABLE CI_BSEG_CL_CHAR ADD CONSTRAINT XT056P0 PRIMARY KEY(BSEG_ID, HEADER_SEQ, SEQNO, CHAR_TYPE_CD) USING INDEX /

Child Table: CI_BSEG_EXCP

CREATE TABLE CI_BSEG_EXCP
{
  BSEG_ID         CHAR(12) NOT NULL ENABLE,
  MESSAGE_CAT_NBR NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,
  MESSAGE_NBR     NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,
  BSEG_EXCP_FLG   CHAR(2) DEFAULT ' ' NOT NULL ENABLE,
  EXP_MSG         VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
  MESSAGE_PARM1   VARCHAR2(2000) DEFAULT ' ' NOT NULL ENABLE,
  MESSAGE_PARM2   VARCHAR2(2000) DEFAULT ' ' NOT NULL ENABLE,
  MESSAGE_PARM3   VARCHAR2(2000) DEFAULT ' ' NOT NULL ENABLE,
  MESSAGE_PARM4   VARCHAR2(2000) DEFAULT ' ' NOT NULL ENABLE,
  MESSAGE_PARM5   VARCHAR2(2000) DEFAULT ' ' NOT NULL ENABLE,
  MESSAGE_PARM6   VARCHAR2(2000) DEFAULT ' ' NOT NULL ENABLE,
  MESSAGE_PARM7   VARCHAR2(2000) DEFAULT ' ' NOT NULL ENABLE,
  MESSAGE_PARM8   VARCHAR2(2000) DEFAULT ' ' NOT NULL ENABLE,
  MESSAGE_PARM9   VARCHAR2(2000) DEFAULT ' ' NOT NULL ENABLE,
  CALL_SEQ        VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
  USER_ID         CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
  CRE_DTTM DATE,
  REVIEW_COMP    CHAR(1) DEFAULT ' ' NOT NULL ENABLE,
  REVIEW_USER_ID CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
  REVIEW_DT DATE,
  COMMENTS VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
  VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
  CONSTRAINT CI_BSEG_EXCP_FK FOREIGN KEY(BSEG_ID) REFERENCES CI_BSEG ON DELETE CASCADE
}
PARTITION BY REFERENCE (CI_BSEG_EXCP_FK)
ENABLE ROW MOVEMENT
/

INDEX
CREATE UNIQUE INDEX XT051P0 ON CI_BSEG_EXCP ( BSEG_ID ) TABLESPACE CM_XT048_IND
GLOBAL PARTITION BY RANGE (BSEG_ID)
Sample SQL for Partitioning with ILM in C2M for CC&B

ALTER TABLE CI_BSEG_EXCP ADD CONSTRAINT XT051P0 PRIMARY KEY(BSEG_ID) USING INDEX /

Child Table: CI_BSEG_MSG
CREATE TABLE CI_BSEG_MSG
(
   BSEG_ID     CHAR(12) NOT NULL ENABLE,
   BILL_MSG_CD CHAR(4) NOT NULL ENABLE,
   VERSION     NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
   CONSTRAINT CI_BSEG_MSG_FK FOREIGN KEY(BSEG_ID) REFERENCES CI_BSEG ON DELETE CASCADE
) PARTITION BY REFERENCE (CI_BSEG_MSG_FK) ENABLE ROW MOVEMENT /

INDEX
CREATE UNIQUE INDEX XT080P0 ON CI_BSEG_MSG (BSEG_ID, BILL_MSG_CD) TABLESPACE CM_XT048_IND
GLOBAL PARTITION BY RANGE (BSEG_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 ( '749999999999' ),
   PARTITION P7 VALUES LESS THAN ( '874999999999' ),
   PARTITION P8 VALUES LESS THAN ( MAXVALUE )
) COMPRESS ADVANCED LOW
/

ALTER TABLE CI_BSEG_MSG ADD CONSTRAINT XT080P0 PRIMARY KEY(BSEG_ID, BILL_MSG_CD) USING INDEX /

Child Table: CI_BSEG_READ
CREATE TABLE CI_BSEG_READ
(
   BSEG_ID           CHAR(12) NOT NULL ENABLE,
   SP_ID             CHAR(10) NOT NULL ENABLE,
   REG_CONST         NUMBER(12,6) DEFAULT 0 NOT NULL ENABLE,
**Parent Table: CI_BSEG**

Sample SQL for Partitioning with ILM in C2M for CC&B D - 16

Oracle Utilities Customer To Meter Database Administrator's Guide

```sql
SEQNO             NUMBER(5,0) NOT NULL ENABLE,
USAGE_FLG         CHAR(2) DEFAULT ' ' NOT NULL ENABLE,
USE_PCT           NUMBER(3,0) DEFAULT 0 NOT NULL ENABLE,
HOW_TO_USE_FLG    CHAR(2) DEFAULT ' ' NOT NULL ENABLE,
MSR_PEAK_QTY_SW   CHAR(1) DEFAULT ' ' NOT NULL ENABLE,
UOM_CD            CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
TOU_CD            CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
START_REG_READ_ID CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
START_READ_DTTM   DATE NOT NULL ENABLE,
START_REG_READING NUMBER(15,6) DEFAULT 0 NOT NULL ENABLE,
END_REG_READ_ID   CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
END_READ_DTTM     DATE NOT NULL ENABLE,
END_REG_READING   NUMBER(15,6) DEFAULT 0 NOT NULL ENABLE,
MSR_QTY           NUMBER(18,6) DEFAULT 0 NOT NULL ENABLE,
FINAL_UOM_CD      CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
FINAL_TOU_CD      CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
FINAL_REG_QTY     NUMBER(18,6) DEFAULT 0 NOT NULL ENABLE,
VERSION           NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
SQI_CD            CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
FINAL_SQI_CD      CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
CONSTRAINT CI_BSEG_READ_FK FOREIGN KEY(BSEG_ID) REFERENCES CI_BSEG ON DELETE CASCADE
)
PARTITION BY REFERENCE (CI_BSEG_READ_FK)
ENABLE ROW MOVEMENT
/

INDEX
CREATE UNIQUE INDEX XT054P0 ON CI_BSEG_READ ( BSEG_ID, SP_ID, SEQNO )
TABLESPACE CM_XT048_IND
GLOBAL PARTITION BY RANGE (BSEG_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 ( '749999999999' ),
PARTITION P7 VALUES LESS THAN ( '874999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
}
COMPRESSION ADVANCED LOW
/

ALTER TABLE CI_BSEG_READ ADD CONSTRAINT XT054P0 PRIMARY KEY(BSEG_ID, SP_ID, SEQNO) USING INDEX
/
CREATE INDEX XT054S1 ON CI_BSEG_READ ( SP_ID ) TABLESPACE CM_XT048_IND /
CREATE INDEX XT054S2 ON CI_BSEG_READ ( START_REG_READ_ID ) TABLESPACE CM_XT048_IND /
CREATE INDEX XT054S3 ON CI_BSEG_READ ( END_REG_READ_ID ) TABLESPACE CM_XT048_IND /

**Child Table: CI_BSEG_SQ**

CREATE TABLE CI_BSEG_SQ

BSEG_ID CHAR(12) NOT NULL ENABLE, 
UOM_CD CHAR(4) NOT NULL ENABLE, 
TOU_CD CHAR(8) NOT NULL ENABLE, 
SQI_CD CHAR(8) NOT NULL ENABLE, 
INIT_SQ NUMBER(18,6) DEFAULT 0 NOT NULL ENABLE, 
BILL_SQ NUMBER(18,6) DEFAULT 0 NOT NULL ENABLE, 
VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE, 
CONSTRAINT CI_BSEG_SQ_FK FOREIGN KEY(BSEG_ID) REFERENCES CI_BSEG ON DELETE CASCADE 
) 
PARTITION BY REFERENCE (CI_BSEG_SQ_FK) 
ENABLE ROW MOVEMENT 
/

INDEX

CREATE UNIQUE INDEX XT055P0 ON CI_BSEG_SQ ( BSEG_ID, UOM_CD, TOU_CD, 
SQI_CD ) TABLESPACE CM_XT048_IND 
GLOBAL PARTITION BY RANGE (BSEG_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 ( '749999999999' ), 
PARTITION P7 VALUES LESS THAN ( '874999999999' ), 
PARTITION P8 VALUES LESS THAN ( MAXVALUE ) 
) 
COMPRESS ADVANCED LOW 
/

ALTER TABLE CI_BSEG_SQ ADD CONSTRAINT XT055P0 PRIMARY KEY(BSEG_ID, 
UOM_CD, TOU_CD, SQI_CD) USING INDEX 
/

Child Table: CI_BSEG_ITEM

CREATE TABLE CI_BSEG_ITEM
(
BSEG_ID CHAR(12) NOT NULL ENABLE, 
SEQNO NUMBER(5,0) NOT NULL ENABLE, 
ITEM_TYPE_CD CHAR(8) DEFAULT ' ' NOT NULL ENABLE, 
ITEM_ID CHAR(10) DEFAULT ' ' NOT NULL ENABLE, 
START_DT DATE NOT NULL ENABLE, 
END_DT DATE NOT NULL ENABLE, 
ITEM_CNT NUMBER(11,2) DEFAULT 0 NOT NULL ENABLE, 
UOM_CD CHAR(4) DEFAULT ' ' NOT NULL ENABLE, 
SVC_QTY NUMBER(18,6) DEFAULT 0 NOT NULL ENABLE, 
VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE, 
CONSTRAINT CI_BSEG_ITEM_FK FOREIGN KEY(BSEG_ID) REFERENCES CI_BSEG ON DELETE CASCADE 
) 
PARTITION BY REFERENCE (CI_BSEG_ITEM_FK) 
ENABLE ROW MOVEMENT 
/

INDEX

CREATE UNIQUE INDEX XT053P0 ON CI_BSEG_ITEM ( BSEG_ID, SEQNO ) 
TABLESPACE CM_XT048_IND 
GLOBAL PARTITION BY RANGE (BSEG_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 ( '749999999999' ),
PARTITION P7 VALUES LESS THAN ( '874999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
}
COMPRESS ADVANCED LOW
/
ALTER TABLE CI_BSEG_ITEM ADD CONSTRAINT XT053P0 PRIMARY KEY(BSEG_ID, SEQNO) USING INDEX
/
Sample SQL for Enabling ILM in C2M for MDM (Initial Install)

This section provides more detail about steps needed to fully support ILM on tables for maintenance objects that support the functionality.

Three maintenance objects are shown:

- **To Do Entry** - does not include a LOB field.
- **Sync Request** - does include a LOB field and has one tablespace per partition.
- **Initial Measurement Data** - includes LOB fields and has one tablespace per subpartition (shown using subretention). Other maintenance object’s implementations can follow the appropriate pattern based on whether there is a LOB field or not.

The following DDL(s):

- Follows Naming convention recommendations for partitions\subpartitions\tablespaces.
- Ensures all the ILM Storage requirements are incorporated, failing which, ILM functionality will not be achieved.
  - Partitions/subpartitions are defined with respective Tablespace.
  - Child Tables are referenced partitioned.
  - Ensures all Compression recommendations are incorporated.

**Maintenance Object: TO DO ENTRY**

**Parent Table: CI_TD_ENTRY**

```sql
CREATE BIGFILE TABLESPACE CM_XT039_P2011JAN DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2011FEB DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2011MAR DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
```
CREATE BIGFILE TABLESPACE CM_XT039_P2011APR DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2011MAY DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2011JUN DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2011JUL DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2011AUG DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2011SEP DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2011OCT DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2011NOV DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_P2011DEC DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_XT039_PMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;

CREATE TABLE CI_TD_ENTRY (
  TD_ENTRY_ID     CHAR(14) NOT NULL ENABLE,
  BATCH_CD        CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
  BATCH_NBR       NUMBER(10,0) DEFAULT 0 NOT NULL ENABLE,
  MESSAGE_CAT_NBR NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,
  MESSAGE_NBR     NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,
  ASSIGNED_TO     CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
  TD_TYPE_CD      CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
  ROLE_ID         CHAR(10) DEFAULT ' ' NOT NULL ENABLE,
  ENTRY_STATUS_FLG CHAR(2) DEFAULT ' ' NOT NULL ENABLE,
  VERSION         NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
  CRE_DTTM DATE,
  ASSIGNED_DTTM DATE,
  COMPLETE_DTTM DATE,
  COMPLETE_USER_ID CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
  COMMENTS        VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
  ASSIGNED_USER_ID CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
  TD_PRIORITY_FLG CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
  ILM_DT DATE,
  ILM_ARCH_SW CHAR(1)
) ENABLE ROW MOVEMENT
PARTITION BY RANGE (ILM_DT)
SUBPARTITION BY RANGE (TD_ENTRY_ID) SUBPARTITION TEMPLATE
{
  SUBPARTITION S01 VALUES LESS THAN ( '12499999999999999999' ),
  SUBPARTITION S02 VALUES LESS THAN ( '24999999999999999999' ),
  SUBPARTITION S03 VALUES LESS THAN ( '37499999999999999999' ),
  SUBPARTITION S04 VALUES LESS THAN ( '49999999999999999999' ),
  SUBPARTITION S05 VALUES LESS THAN ( '62499999999999999999' ),
  SUBPARTITION S06 VALUES LESS THAN ( '74999999999999999999' ),
  SUBPARTITION S07 VALUES LESS THAN ( '87499999999999999999' ),
}
SAMPLE SQL FOR ENABLING ILM IN C2M FOR MDM (INITIAL INSTALL)

CREATE BIGFILE TABLESPACE CM_XT039_IND DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;

CREATE UNIQUE INDEX XT039P0 ON CI_TD_ENTRY (TD_ENTRY_ID) TABLESPACE CM_XT039_IND GLOBAL PARTITION BY RANGE (TD_ENTRY_ID) WITH NO OVERLAP (PARTITION P1 VALUES LESS THAN ( '12499999999999' ), PARTITION P2 VALUES LESS THAN ( '24999999999999' ), PARTITION P3 VALUES LESS THAN ( '37499999999999' ), PARTITION P4 VALUES LESS THAN ( '49999999999999' ), PARTITION P5 VALUES LESS THAN ( '62499999999999' ), PARTITION P6 VALUES LESS THAN ( '74999999999999' ), PARTITION P7 VALUES LESS THAN ( '87499999999999' ), PARTITION P8 VALUES LESS THAN ( MAXVALUE ), PARTITION PMAX VALUES LESS THAN ( MAXVALUE ) TABLESPACE CM_XT039_PMAX);

ALTER TABLE CI_TD_ENTRY ADD CONSTRAINT XT039P0 PRIMARY KEY (TD_ENTRY_ID) USING INDEX;
CREATE UNIQUE INDEX XT039S2 ON CI_TD_ENTRY (ASSIGNED_TO, TD_ENTRY_ID) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

CREATE INDEX XT039S3 ON CI_TD_ENTRY (ENTRY_STATUS_FLG, ASSIGNED_TO) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

CREATE INDEX XT039S4 ON CI_TD_ENTRY (ROLE_ID, TD_TYPE_CD, ENTRY_STATUS_FLG, TD_PRIORITY_FLG) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

CREATE INDEX XT039S5 ON CI_TD_ENTRY (BATCH_CD, BATCH_NBR, ENTRY_STATUS_FLG) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

CREATE UNIQUE INDEX XT039S6 ON CI_TD_ENTRY (TD_ENTRY_ID, ASSIGNED_TO, ENTRY_STATUS_FLG) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

CREATE UNIQUE INDEX XT039S7 ON CI_TD_ENTRY (COMPLETE_USER_ID, COMPLETE_DTTM, TD_ENTRY_ID) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

CREATE UNIQUE INDEX CM_ILM_XT039S8 ON CI_TD_ENTRY (ILM_DT, ILM_ARCH_SW, TD_ENTRY_ID) LOCAL COMPRESS ADVANCED LOW;

CREATE TABLE CI_TD_DRLKEY
(
TD_ENTRY_ID CHAR(14) 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
ON DELETE CASCADE)
PARTITION BY REFERENCE (CI_TD_DRLKEY_FK)
ENABLE ROW MOVEMENT;

INDEX

CREATE UNIQUE INDEX XT037P0 ON CI_TD_DRLKEY (TD_ENTRY_ID, SEQ_NUM) TABLESPACE CM_XT039_IND
GLOBAL PARTITION BY RANGE (TD_ENTRY_ID)
(
PARTITION P1 VALUES LESS THAN ( '12499999999999' ),
PARTITION P2 VALUES LESS THAN ( '24999999999999' ),
PARTITION P3 VALUES LESS THAN ( '37499999999999' ),
PARTITION P4 VALUES LESS THAN ( '49999999999999' ),
PARTITION P5 VALUES LESS THAN ( '62499999999999' ),
PARTITION P6 VALUES LESS THAN ( '74999999999999' ),
PARTITION P7 VALUES LESS THAN ( '87499999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
)
COMPRESS ADVANCED LOW;

ALTER TABLE CI_TD_DRLKEY ADD CONSTRAINT XT037P0 PRIMARY KEY(TD_ENTRY_ID, SEQ_NUM) USING INDEX;

CREATE INDEX XT037S1 ON CI_TD_DRLKEY (KEY_VALUE, TD_ENTRY_ID) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;
CREATE TABLE CI_TD_ENTRY_CHA
(
    TD_ENTRY_ID CHAR(14) NOT NULL ENABLE,
    CHAR_TYPE_CD CHAR(8) NOT NULL ENABLE,
    SEQ_NUM NUMBER(3,0) DEFAULT 0 NOT NULL ENABLE,
    CHAR_VAL CHAR(16) DEFAULT ' ' NOT NULL ENABLE,
    VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    ADHOC_CHAR_VAL VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK1 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK2 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK3 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK4 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK5 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    SRCH_CHAR_VAL VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    CONSTRAINT CI_TD_ENTRY_CHA_FK FOREIGN KEY(TD_ENTRY_ID) REFERENCES CI_TD_ENTRY
    ON DELETE CASCADE)
PARTITION BY REFERENCE (CI_TD_ENTRY_CHA_FK)
ENABLE ROW MOVEMENT;

INDEX
CREATE UNIQUE INDEX XT701P0 ON CI_TD_ENTRY_CHA (TD_ENTRY_ID,
CHAR_TYPE_CD, SEQ_NUM) TABLESPACE CM_XT039_IND
GLOBAL PARTITION BY RANGE (TD_ENTRY_ID)
(
    PARTITION P1 VALUES LESS THAN ('12499999999999'),
    PARTITION P2 VALUES LESS THAN ('24999999999999'),
    PARTITION P3 VALUES LESS THAN ('37499999999999'),
    PARTITION P4 VALUES LESS THAN ('49999999999999'),
    PARTITION P5 VALUES LESS THAN ('62499999999999'),
    PARTITION P6 VALUES LESS THAN ('74999999999999'),
    PARTITION P7 VALUES LESS THAN ('87499999999999'),
    PARTITION P8 VALUES LESS THAN (MAXVALUE)
)
COMPRESS ADVANCED LOW;

ALTER TABLE CI_TD_ENTRY_CHA ADD CONSTRAINT XT701P0 PRIMARY KEY(TD_ENTRY_ID, CHAR_TYPE_CD, SEQ_NUM) USING INDEX;

CREATE INDEX XT701S1 ON CI_TD_ENTRY_CHA (SRCH_CHAR_VAL, CHAR_TYPE_CD,
TD_ENTRY_ID) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

CREATE INDEX XT701S2 ON CI_TD_ENTRY_CHA (CHAR_VAL_FK1) TABLESPACE
CM_XT039_IND
COMPRESS ADVANCED LOW;

CREATE TABLE CI_TD_LOG
(
    TD_ENTRY_ID CHAR(14) NOT NULL ENABLE,
    SEQ_NUM NUMBER(3,0) NOT NULL ENABLE,
    LOG_DTTM DATE NOT NULL ENABLE,
    LOG_TYPE_FLG CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
    USER_ID CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
    ASSIGNED_TO CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
    VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
Sample SQL for Enabling ILM in C2M for MDM (Initial Install)

CREATE TABLE CI_TD_MSG_PARM
(
  TD_ENTRY_ID CHAR(14) NOT NULL ENABLE,
  SEQ_NUM NUMBER(3,0) NOT NULL ENABLE,
  MSG_PARM_VAL VARCHAR2(2000) 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 ON DELETE CASCADE
)
PARTITION BY REFERENCE (CI_TD_MSG_PARM_FK)
ENABLE ROW MOVEMENT;

INDEX
CREATE UNIQUE INDEX XT040P0 ON CI_TD_MSG_PARM ( TD_ENTRY_ID, SEQ_NUM )
TABLESPACE CM_XT039_IND
GLOBAL PARTITION BY RANGE (TD_ENTRY_ID)
(
  PARTITION P1 VALUES LESS THAN ( '12499999999999' ),
  PARTITION P2 VALUES LESS THAN ( '24999999999999' ),
  PARTITION P3 VALUES LESS THAN ( '37499999999999' ),
  PARTITION P4 VALUES LESS THAN ( '49999999999999' ),
  PARTITION P5 VALUES LESS THAN ( '62499999999999' ),
  PARTITION P6 VALUES LESS THAN ( '74999999999999' ),
  PARTITION P7 VALUES LESS THAN ( '87499999999999' ),
  PARTITION P8 VALUES LESS THAN ( MAXVALUE )
)
COMPRESS ADVANCED LOW;

ALTER TABLE CI_TD_MSG_PARM ADD CONSTRAINT XT040P0 PRIMARY KEY(TD_ENTRY_ID, SEQ_NUM) USING INDEX;

CREATE INDEX XT040S1 ON CI_TD_MSG_PARM ( LOG_DTTM, USER_ID, LOG_TYPE_FLG, TD_ENTRY_ID ) TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

Child Table: CI_TD_MSG_PARM

CREATE TABLE CI_TD_MSG_PARM
(
  TD_ENTRY_ID CHAR(14) NOT NULL ENABLE,
  SEQ_NUM NUMBER(3,0) NOT NULL ENABLE,
  MSG_PARM_VAL VARCHAR2(2000) 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 ON DELETE CASCADE
)
PARTITION BY REFERENCE (CI_TD_MSG_PARM_FK)
ENABLE ROW MOVEMENT;

INDEX
CREATE UNIQUE INDEX XT040P0 ON CI_TD_MSG_PARM ( TD_ENTRY_ID, SEQ_NUM )
TABLESPACE CM_XT039_IND
GLOBAL PARTITION BY RANGE (TD_ENTRY_ID)
(
  PARTITION P1 VALUES LESS THAN ( '12499999999999' ),
  PARTITION P2 VALUES LESS THAN ( '24999999999999' ),
  PARTITION P3 VALUES LESS THAN ( '37499999999999' ),
  PARTITION P4 VALUES LESS THAN ( '49999999999999' ),
  PARTITION P5 VALUES LESS THAN ( '62499999999999' ),
  PARTITION P6 VALUES LESS THAN ( '74999999999999' ),
  PARTITION P7 VALUES LESS THAN ( '87499999999999' ),
  PARTITION P8 VALUES LESS THAN ( MAXVALUE )
)
COMPRESS ADVANCED LOW;
ALTER TABLE CI_TD_MSG_PARM ADD CONSTRAINT XT040P0 PRIMARY KEY(TD_ENTRY_ID, SEQ_NUM) USING INDEX;

Child Table: CI_TD_SRTKEY

CREATE TABLE CI_TD_SRTKEY
(
  TD_ENTRY_ID CHAR(14) 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 ON DELETE CASCADE
)
PARTITION BY REFERENCE (CI_TD_SRTKEY_FK)
ENABLE ROW MOVEMENT;

INDEX
CREATE UNIQUE INDEX XT041P0 ON CI_TD_SRTKEY ( TD_ENTRY_ID, SEQ_NUM )
TABLESPACE CM_XT039_IND
GLOBAL PARTITION BY RANGE (TD_ENTRY_ID)
(
  PARTITION P1 VALUES LESS THAN ( '12499999999999' ),
  PARTITION P2 VALUES LESS THAN ( '24999999999999' ),
  PARTITION P3 VALUES LESS THAN ( '37499999999999' ),
  PARTITION P4 VALUES LESS THAN ( '49999999999999' ),
  PARTITION P5 VALUES LESS THAN ( '62499999999999' ),
  PARTITION P6 VALUES LESS THAN ( '74999999999999' ),
  PARTITION P7 VALUES LESS THAN ( '87499999999999' ),
  PARTITION P8 VALUES LESS THAN ( MAXVALUE )
)
COMpress ADVANCED LOW;

ALTER TABLE CI_TD_SRTKEY ADD CONSTRAINT XT041P0 PRIMARY KEY(TD_ENTRY_ID, SEQ_NUM) USING INDEX;

CREATE INDEX XT041S1 ON CI_TD_SRTKEY ( KEY_VALUE, TD_ENTRY_ID )
TABLESPACE CM_XT039_IND COMPRESS ADVANCED LOW;

Maintenance Object:F1-SYNCREQIN

Parent Table: F1_SYNC_REQ_IN

CREATE BIGFILE TABLESPACE CM_F1T191_P2011JAN DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2011FEB DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2011MAR DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2011APR DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2011MAY DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2011JUN DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2011JUL DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2011AUG DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2011SEP DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2011OCT DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2011NOV DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_P2011DEC DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_F1T191_PMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;

CREATE TABLE F1_SYNC_REQ_IN
(
    F1_SYNC_REQ_IN_ID CHAR(14) NOT NULL ENABLE,
    BUS_OBJ_CD        CHAR(30) DEFAULT ' ' NOT NULL ENABLE,
    CRE_DTTM DATE NOT NULL ENABLE,
    BO_STATUS_CD CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
    STATUS_UPD_DTTM DATE,
    MAINT_OBJ_CD  CHAR(12 BYTE) DEFAULT ' ' NOT NULL ENABLE,
    NT_XID_CD     CHAR(30 BYTE) DEFAULT ' ' NOT NULL ENABLE,
    EXT_PK_VALUE1 VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    EXT_PK_VALUE2 VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    EXT_PK_VALUE3 VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    EXT_PK_VALUE4 VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    EXT_PK_VALUE5 VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    PK_VALUE1     VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    BO_DATA_AREA CLOB,
    PRE_TRN_INIT_BO_DATA_AREA CLOB,
    PRE_TRN_FIN_BO_DATA_AREA CLOB,
    POST_TRN_BO_DATA_AREA CLOB,
    VERSION                  NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    EXT_REFERENCE_ID         CHAR(36) DEFAULT ' ' NOT NULL ENABLE,
    F1_INITIAL_LOAD_SYNC_FLG CHAR(14) DEFAULT ' ' NOT NULL ENABLE,
    F1_COMPOSITE_SYNC_FLG    CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
    ILM_DT DATE,
    ILM_ARCH_SW CHAR(1)
)

ENABLE ROW MOVEMENT
LOB (BO_DATA_AREA)   STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
LOB (PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
LOB (PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
LOB (POST_TRN_BO_DATA_AREA)   STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
PARTITION BY RANGE(ILM_DT)
  SUBPARTITION BY RANGE(F1_SYNC_REQ_IN_ID)
  SUBPARTITION TEMPLATE
    {
      SUBPARTITION S01 VALUES LESS THAN ( '12499999999999' ),
      SUBPARTITION S02 VALUES LESS THAN ( '24999999999999' ),
      SUBPARTITION S03 VALUES LESS THAN ( '37499999999999' ),
      SUBPARTITION S04 VALUES LESS THAN ( '49999999999999' ),
      SUBPARTITION S05 VALUES LESS THAN ( '62499999999999' ),
      SUBPARTITION S06 VALUES LESS THAN ( '74999999999999' ),
      SUBPARTITION S07 VALUES LESS THAN ( '87499999999999' ),
      SUBPARTITION SMAX VALUES LESS THAN ( MAXVALUE )
    }
  PARTITION "P2011JAN" VALUES LESS THAN (TO_DATE('2011-02-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011JAN )
    LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011JAN )
    LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011JAN )
    TABLESPACE_CM_F1T191_P2011JAN,
  PARTITION "P2011FEB" VALUES LESS THAN (TO_DATE('2011-03-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011FEB )
    LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011FEB )
    LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011FEB )
    LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011FEB )
    TABLESPACE_CM_F1T191_P2011FEB,
  PARTITION "P2011MAR" VALUES LESS THAN (TO_DATE('2011-04-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011MAR )
    LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011MAR )
    LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011MAR )
    LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011MAR )
    TABLESPACE_CM_F1T191_P2011MAR,
  PARTITION "P2011APR" VALUES LESS THAN (TO_DATE('2011-05-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011APR )
    LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011APR )
    LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011APR )
    LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011APR )
    TABLESPACE_CM_F1T191_P2011APR,
  PARTITION "P2011MAY" VALUES LESS THAN (TO_DATE('2011-06-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011MAY )
Sample SQL for Enabling ILM in C2M for MDM (Initial Install)

```sql
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011MAY )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011MAY )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011MAY )
TABLESPACE CM_F1T191_P2011MAY,
PARTITION "P2011JUN" VALUES LESS THAN (TO_DATE('2011-07-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011MAY )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011MAY )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011MAY )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011MAY )
TABLESPACE CM_F1T191_P2011MAY,
PARTITION "P2011JUL" VALUES LESS THAN (TO_DATE('2011-08-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011JUL )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011JUL )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011JUL )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011JUL )
TABLESPACE CM_F1T191_P2011JUL,
PARTITION "P2011AUG" VALUES LESS THAN (TO_DATE('2011-09-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011AUG )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011AUG )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011AUG )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011AUG )
TABLESPACE CM_F1T191_P2011AUG,
PARTITION "P2011SEP" VALUES LESS THAN (TO_DATE('2011-10-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011SEP )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011SEP )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011SEP )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011SEP )
TABLESPACE CM_F1T191_P2011SEP,
PARTITION "P2011OCT" VALUES LESS THAN (TO_DATE('2011-11-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011OCT )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011OCT )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011OCT )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011OCT )
TABLESPACE CM_F1T191_P2011OCT,
```
PARTITION "P2011NOV" VALUES LESS THAN (TO_DATE('2011-12-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011NOV )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011NOV )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011NOV )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011NOV )
TABLESPACE CM_F1T191_P2011NOV,

PARTITION "P2011DEC" VALUES LESS THAN (TO_DATE('2012-01-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011DEC )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011DEC )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011DEC )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011DEC )
TABLESPACE CM_F1T191_P2011DEC,

PARTITION "PMAX" VALUES LESS THAN (MAXVALUE)
LOB(BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_PMAX )
LOB(PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_PMAX )
LOB(PRE_TRN_FIN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_PMAX )
LOB(POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_PMAX )
TABLESPACE CM_F1T191_PMAX
);

INDEX
CREATE BIGFILE TABLESPACE CM_F1T191_IND DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;

CREATE UNIQUE INDEX F1T191P0 ON F1_SYNC_REQ_IN(F1_SYNC_REQ_IN_ID) TABLESPACE CM_F1T191_IND GLOBAL PARTITION BY RANGE (F1_SYNC_REQ_IN_ID) { PARTITION P1 VALUES LESS THAN ( '12499999999999' ), PARTITION P2 VALUES LESS THAN ( '24999999999999' ), PARTITION P3 VALUES LESS THAN ( '37499999999999' ), PARTITION P4 VALUES LESS THAN ( '49999999999999' ), PARTITION P5 VALUES LESS THAN ( '62499999999999' ), PARTITION P6 VALUES LESS THAN ( '74999999999999' ), PARTITION P7 VALUES LESS THAN ( '87499999999999' ), PARTITION P8 VALUES LESS THAN ( MAXVALUE ) };

ALTER TABLE F1_SYNC_REQ_IN ADD CONSTRAINT F1T191P0 PRIMARY KEY (F1_SYNC_REQ_IN_ID) USING INDEX;

CREATE UNIQUE INDEX F1T191S1 ON F1_SYNC_REQ_IN (BO_STATUS_CD, BUS_OBJ_CD, F1_SYNC_REQ_IN_ID) TABLESPACE CM_F1T191_IND COMPRESS ADVANCED LOW;
CREATE INDEX F1T191S2 ON F1_SYNC_REQ_IN(MAINT_OBJ_CD, Ext_PK_VALUE1, NT_XID_CD, PK_VALUE1) TABLESPACE CM_F1T191_IND COMPRESS ADVANCED LOW;

CREATE INDEX F1T191S3 ON F1_SYNC_REQ_IN(Ext_REFERENCE_ID) TABLESPACE CM_F1T191_IND;
CREATE UNIQUE INDEX CM_ILM_F1T191S3 ON F1_SYNC_REQ_IN(ILM_DT, ILM_ARCH_SW, F1_SYNC_REQ_IN_ID) LOCAL COMPRESS ADVANCED LOW;

Child Table: F1_SYNC_REQ_IN_CHAR

CREATE TABLE F1_SYNC_REQ_IN_CHAR
(
  F1_SYNC_REQ_IN_ID CHAR(14) NOT NULL ENABLE,
  CHAR_TYPE_CD CHAR(8) NOT NULL ENABLE,
  SEQ_NUM NUMBER(3,0) NOT NULL ENABLE,
  CHAR_Val CHAR(16) DEFAULT ' ' NOT NULL ENABLE,
  ADHOC_CHAR_VAL VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK1 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK2 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK3 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK4 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK5 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  SRCH_CHAR_VAL VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
  CONSTRAINT F1_SYNC_REQ_IN_CHAR_FK FOREIGN KEY (F1_SYNC_REQ_IN_ID)
  REFERENCES F1_SYNC_REQ_IN ON DELETE CASCADE)
PARTITION BY REFERENCE (F1_SYNC_REQ_IN_CHAR_FK)
ENABLE ROW MOVEMENT;

INDEX
CREATE UNIQUE INDEX F1T193P0 ON F1_SYNC_REQ_IN_CHAR(F1_SYNC_REQ_IN_ID, CHAR_TYPE_CD, SEQ_NUM) TABLESPACE CM_F1T191_IND
GLOBAL PARTITION BY RANGE (F1_SYNC_REQ_IN_ID)
(
  PARTITION P1 VALUES LESS THAN ( '12499999999999' ),
  PARTITION P2 VALUES LESS THAN ( '24999999999999' ),
  PARTITION P3 VALUES LESS THAN ( '37499999999999' ),
  PARTITION P4 VALUES LESS THAN ( '49999999999999' ),
  PARTITION P5 VALUES LESS THAN ( '62499999999999' ),
  PARTITION P6 VALUES LESS THAN ( '74999999999999' ),
  PARTITION P7 VALUES LESS THAN ( '87499999999999' ),
  PARTITION P8 VALUES LESS THAN ( MAXVALUE )
)
COMPRESS ADVANCED LOW;

ALTER TABLE F1_SYNC_REQ_IN_CHAR ADD CONSTRAINT F1T193P0 PRIMARY KEY
(F1_SYNC_REQ_IN_ID, CHAR_TYPE_CD, SEQ_NUM) USING INDEX;
CREATE INDEX F1T193S1 ON F1_SYNC_REQ_IN_CHAR(SRCH_CHAR_VAL) TABLESPACE CM_F1T191_IND ;

Child Table: F1_SYNC_REQ_IN_EXCP

CREATE TABLE F1_SYNC_REQ_IN_EXCP
(
  F1_SYNC_REQ_IN_ID CHAR(14) NOT NULL ENABLE,
  SEQNO NUMBER(5,0) NOT NULL ENABLE,
CREATE TABLE F1_SYNC_REQ_IN_EXCP_PARM
(
  F1_SYNC_REQ_IN_ID CHAR(14) NOT NULL ENABLE,
  SEQNO NUMBER(5,0) NOT NULL ENABLE,
  PARM_SEQ NUMBER(3,0) NOT NULL ENABLE,
  MSG_PARM_VAL VARCHAR2(2000) DEFAULT ' ' NOT NULL ENABLE,
  MSG_PARM_TYP_FLG CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
  VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
  CONSTRAINT F1_SYNC_REQ_IN_EXCP_PARM_FK FOREIGN KEY(F1_SYNC_REQ_IN_ID)
  REFERENCES F1_SYNC_REQ_IN ON DELETE CASCADE)
PARTITION BY REFERENCE (F1_SYNC_REQ_IN_EXCP_PARM_FK)
ENABLE ROW MOVEMENT;

INDEX
CREATE UNIQUE INDEX F1T198P0 ON
F1_SYNC_REQ_IN_EXCP_PARM(F1_SYNC_REQ_IN_ID,SEQNO,PARM_SEQ) TABLESPACE CM_F1T191_IND
GLOBAL PARTITION BY RANGE (F1_SYNC_REQ_IN_ID)
(
  PARTITION P1 VALUES LESS THAN ( '12499999999999' ),
  PARTITION P2 VALUES LESS THAN ( '24999999999999' ),
  PARTITION P3 VALUES LESS THAN ( '37499999999999' ),
  PARTITION P4 VALUES LESS THAN ( '49999999999999' ),
  PARTITION P5 VALUES LESS THAN ( '62499999999999' ),
  PARTITION P6 VALUES LESS THAN ( '74999999999999' ),
  PARTITION P7 VALUES LESS THAN ( '87499999999999' ),
  PARTITION P8 VALUES LESS THAN ( MAXVALUE )
)
COMpress ADVANCED LOW;

ALTER TABLE F1_SYNC_REQ_IN_EXCP ADD CONSTRAINT F1T198P0 PRIMARY KEY
(F1_SYNC_REQ_IN_ID,SEQNO,PARM_SEQ) USING INDEX;

Sample SQL for Enabling ILM in C2M for MDM (Initial Install) E - 13
Oracle Utilities Customer To Meter Database Administrator’s Guide
Child Table: F1_SYNC_REQ_IN_LOG

```sql
CREATE TABLE F1_SYNC_REQ_IN_LOG
(
    F1_SYNC_REQ_IN_ID CHAR(14) NOT NULL ENABLE,
    SEQNO NUMBER(5,0) NOT NULL ENABLE,
    LOG_ENTRY_TYPE_FLG CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
    LOG_DTTM DATE NOT NULL ENABLE,
    BO_STATUS_CD CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
    MESSAGE_CAT_NBR NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,
    MESSAGE_NBR NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,
    CHAR_TYPE_CD CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL CHAR(16) DEFAULT ' ' NOT NULL ENABLE,
    ADHOC_CHAR_VAL VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK1 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK2 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK3 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK4 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK5 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
    DESCRLONG VARCHAR2(4000) DEFAULT ' ' NOT NULL ENABLE,
    USER_ID CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
    VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    CONSTRAINT F1_SYNC_REQ_IN_LOG_FK FOREIGN KEY (F1_SYNC_REQ_IN_ID)
        REFERENCES F1_SYNC_REQ_IN ON DELETE CASCADE
) PARTITION BY REFERENCE (F1_SYNC_REQ_IN_LOG_FK)
ENABLE ROW MOVEMENT;
```

INDEX

```sql
CREATE UNIQUE INDEX F1T194P0 ON
F1_SYNC_REQ_IN_LOG(F1_SYNC_REQ_IN_ID,SEQNO) TABLESPACE CM_F1T191_IND
GLOBAL PARTITION BY RANGE (F1_SYNC_REQ_IN_ID)
(
    PARTITION P1 VALUES LESS THAN ( '12499999999999' ),
    PARTITION P2 VALUES LESS THAN ( '24999999999999' ),
    PARTITION P3 VALUES LESS THAN ( '37499999999999' ),
    PARTITION P4 VALUES LESS THAN ( '49999999999999' ),
    PARTITION P5 VALUES LESS THAN ( '62499999999999' ),
    PARTITION P6 VALUES LESS THAN ( '74999999999999' ),
    PARTITION P7 VALUES LESS THAN ( '87499999999999' ),
    PARTITION P8 VALUES LESS THAN ( MAXVALUE )
) COMPRESS ADVANCED LOW;
```

Sample SQL for Enabling ILM in C2M for MDM (Initial Install) E - 14
Oracle Utilities Customer To Meter Database Administrator’s Guide
### Child Table: F1_SYNC_REQ_IN_LOG_PARM

```sql
CREATE TABLE F1_SYNC_REQ_IN_LOG_PARM
(
    F1_SYNC_REQ_IN_ID CHAR(14) NOT NULL ENABLE,
    SEQNO NUMBER(5,0) NOT NULL ENABLE,
    PARM_SEQ NUMBER(3,0) NOT NULL ENABLE,
    MSG_PARM_VAL VARCHAR2(2000) DEFAULT ' ' NOT NULL ENABLE,
    MSG_PARM_TYP_FLG CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
    VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    CONSTRAINT F1_SYNC_REQ_IN_LOG_PARM_FK FOREIGN KEY(F1_SYNC_REQ_IN_ID) REFERENCES F1_SYNC_REQ_IN ON DELETE CASCADE
) PARTITION BY REFERENCE (F1_SYNC_REQ_IN_LOG_PARM_FK) ENABLE ROW MOVEMENT;
```

### INDEX

```sql
CREATE UNIQUE INDEX F1T195P0 ON
F1_SYNC_REQ_IN_LOG_PARM(F1_SYNC_REQ_IN_ID,SEQNO,PARM_SEQ) TABLESPACE CM_F1T191_IND
GLOBAL PARTITION BY RANGE (F1_SYNC_REQ_IN_ID)
(
    PARTITION P1 VALUES LESS THAN ( '12499999999999' ),
    PARTITION P2 VALUES LESS THAN ( '24999999999999' ),
    PARTITION P3 VALUES LESS THAN ( '37499999999999' ),
    PARTITION P4 VALUES LESS THAN ( '49999999999999' ),
    PARTITION P5 VALUES LESS THAN ( '62499999999999' ),
    PARTITION P6 VALUES LESS THAN ( '74999999999999' ),
    PARTITION P7 VALUES LESS THAN ( '87499999999999' ),
    PARTITION P8 VALUES LESS THAN ( MAXVALUE )
) COMPRESSION ADVANCED LOW;
```

```sql
ALTER TABLE F1_SYNC_REQ_IN_LOG_PARM ADD CONSTRAINT F1T195P0 PRIMARY KEY (F1_SYNC_REQ_IN_ID,SEQNO,PARM_SEQ) USING INDEX;
```

### Child Table: F1_SYNC_REQ_IN_REL_OBJ

```sql
CREATE TABLE F1_SYNC_REQ_IN_REL_OBJ
(
    F1_SYNC_REQ_IN_ID CHAR(14) NOT NULL ENABLE,
    MAINT_OBJ_CD CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
    REL_OBJ_TYPE_FLG CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
    PK_VALUE1 VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    PK_VALUE2 VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    PK_VALUE3 VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    PK_VALUE4 VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    PK_VALUE5 VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
    VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    CONSTRAINT F1_SYNC_REQ_IN_REL_OBJ_FK FOREIGN KEY(F1_SYNC_REQ_IN_ID) REFERENCES F1_SYNC_REQ_IN_REL_OBJ_FK ON DELETE CASCADE)
PARTITION BY REFERENCE (F1_SYNC_REQ_IN_REL_OBJ_FK) ENABLE ROW MOVEMENT;
```

### INDEX

```sql
CREATE UNIQUE INDEX F1T192P0 ON
F1_SYNC_REQ_IN_REL_OBJ(F1_SYNC_REQ_IN_ID, MAINT_OBJ_CD, REL_OBJ_TYPE_FLG) TABLESPACE CM_F1T191_IND
```
GLOBAL PARTITION BY RANGE (F1_SYNC_REQ_IN_ID) 
{
PARTITION P1 VALUES LESS THAN ( '1249999999999999' ),
PARTITION P2 VALUES LESS THAN ( '2499999999999999' ),
PARTITION P3 VALUES LESS THAN ( '3749999999999999' ),
PARTITION P4 VALUES LESS THAN ( '4999999999999999' ),
PARTITION P5 VALUES LESS THAN ( '6249999999999999' ),
PARTITION P6 VALUES LESS THAN ( '7499999999999999' ),
PARTITION P7 VALUES LESS THAN ( '8749999999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
} COMPRESS ADVANCED LOW;

ALTER TABLE F1_SYNC_REQ_IN_REL_OBJ ADD CONSTRAINT F1T192P0 PRIMARY KEY 
(F1_SYNC_REQ_IN_ID, MAINT_OBJ_CD, REL_OBJ_TYPE_FLG) USING INDEX;

CREATE INDEX F1T192S1 ON F1_SYNC_REQ_IN_REL_OBJ(PK_VALUE1) TABLESPACE 
CM_F1T191_IND;

Maintenance Object: D1-IMD

Parent Table: D1_INIT_MSRMT_DATA

CREATE BIGFILE TABLESPACE CM_D1T304_P2011JAN_S181 DATAFILE '+DATADG' 
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS 
ADVANCED;

CREATE BIGFILE TABLESPACE CM_D1T304_P2011JAN_SMAX DATAFILE '+DATADG' 
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS 
ADVANCED;

CREATE BIGFILE TABLESPACE CM_D1T304_P2011FEB_S181 DATAFILE '+DATADG' 
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS 
ADVANCED;

CREATE BIGFILE TABLESPACE CM_D1T304_P2011FEB_SMAX DATAFILE '+DATADG' 
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS 
ADVANCED;

CREATE BIGFILE TABLESPACE CM_D1T304_P2011MAR_S181 DATAFILE '+DATADG' 
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS 
ADVANCED;

CREATE BIGFILE TABLESPACE CM_D1T304_P2011MAR_SMAX DATAFILE '+DATADG' 
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS 
ADVANCED;

CREATE BIGFILE TABLESPACE CM_D1T304_P2011APR_S181 DATAFILE '+DATADG' 
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS 
ADVANCED;

CREATE BIGFILE TABLESPACE CM_D1T304_P2011APR_SMAX DATAFILE '+DATADG' 
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS 
ADVANCED;

CREATE BIGFILE TABLESPACE CM_D1T304_P2011MAY_S181 DATAFILE '+DATADG' 
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS 
ADVANCED;

CREATE BIGFILE TABLESPACE CM_D1T304_P2011MAY_SMAX DATAFILE '+DATADG' 
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS 
ADVANCED;

CREATE BIGFILE TABLESPACE CM_D1T304_P2011JUN_S181 DATAFILE '+DATADG' 
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS 
ADVANCED;
CREATE TABLE D1_INIT_MSRMT_DATA
(
    INIT_MSRMT_DATA_ID CHAR(14) NOT NULL ENABLE,
    MEASR_COMP_ID      CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
    D1_FROM_DTTM DATE,
    D1_TO_DTTM DATE,
    DATA_SRC_FLG        CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
    TIME_ZONE_CD        CHAR(10) DEFAULT ' ' NOT NULL ENABLE,
    BUS_OBJ_CD          CHAR(30) DEFAULT ' ' NOT NULL ENABLE,
    BO_STATUS_CD        CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
    BO_STATUS_REASON_CD VARCHAR2(30) DEFAULT ' ' NOT NULL ENABLE,
    IMD_BO_DATA_AREA CLOB,
    STATUS_UPD_DTTM DATE NOT NULL ENABLE,
    CRE_DTTM DATE NOT NULL ENABLE,
    VERSION    NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    IMD_EXT_ID VARCHAR2(120),
    ...
CREATE TABLE D1_INIT_MSRMT_DATA
(PREVEE_BO_DATA_AREA CLOB,
POSTVEE_BO_DATA_AREA CLOB,
TRACE_BO_DATA_AREA CLOB,
RAW_BO_DATA_AREA CLOB,
LAST_UPDATE_DTTM DATE,
ILM_DT DATE,
ILM_ARCH_SW CHAR(1),
RETENTION_PERIOD NUMBER(5,0) DEFAULT 99999 NOT NULL ENABLE
)
ENABLE ROW MOVEMENT PCTFREE 50
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
PARTITION BY RANGE (ILM_DT)
SUBPARTITION BY range (RETENTION_PERIOD)
(
PARTITION "P2011JAN" VALUES LESS THAN (TO_DATE('2011-02-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
SUBPARTITION P2011JAN_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011JAN_S181

LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_S181)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_S181)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_S181)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_S181)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_S181)
)

SUBPARTITION P2011JAN_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011JAN_SMAX

LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_SMAX)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_SMAX)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_SMAX)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_SMAX)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_SMAX)
)
,
PARTITION "P2011FEB" VALUES LESS THAN (TO_DATE('2011-03-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
SUBPARTITION P2011FEB_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011FEB_S181

LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)
);
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)
    LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)

SUBPARTITION P2011FEB_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011FEB_SMAX
    LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_SMAX)
    LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_SMAX)
    LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_SMAX)
    LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_SMAX)
    LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_SMAX)

PARTITION "P2011MAR" VALUES LESS THAN (TO_DATE('2011-04-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))

SUBPARTITION P2011MAR_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011MAR_S181
    LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_S181)
    LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_S181)
    LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_S181)
    LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_S181)
    LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_S181)

SUBPARTITION P2011MAR_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011MAR_SMAX
    LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_SMAX)
    LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_SMAX)
    LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_SMAX)
    LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_SMAX)
    LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_SMAX)

PARTITION "P2011APR" VALUES LESS THAN (TO_DATE('2011-05-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))

SUBPARTITION P2011APR_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011APR_S181
    LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_S181)
    LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_S181)
    LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_S181)
    LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_S181)
    LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_S181)
SUBPARTITION P2011APR_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011APR_SMAX
  LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_SMAX)
  LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_SMAX)
  LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_SMAX)
  LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_SMAX)
  LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_SMAX)
)
PARTITION "P2011MAY" VALUES LESS THAN (TO_DATE('2011-06-01 00:00:01',
  'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))(
  SUBPARTITION P2011MAY_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011MAY_S181
  LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_S181)
  LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_S181)
  LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_S181)
  LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_S181)
  LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_S181)
)
  SUBPARTITION P2011MAY_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011MAY_SMAX
  LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_SMAX)
  LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_SMAX)
  LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_SMAX)
  LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_SMAX)
  LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_SMAX)
),
PARTITION "P2011JUN" VALUES LESS THAN (TO_DATE('2011-07-01 00:00:01',
  'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))(
  SUBPARTITION P2011JUN_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011JUN_S181
  LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_S181)
  LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_S181)
  LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_S181)
  LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_S181)
  LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_S181)
)
  SUBPARTITION P2011JUN_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011JUN_SMAX
  LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
  LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
  LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
  LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
  LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
  LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
  LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
),
PARTITION "P2011JUL" VALUES LESS THAN (TO_DATE('2011-08-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))( 
  SUBPARTITION P2011JUL_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011JUL_S181
    LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_S181)
    LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_S181)
    LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_S181)
    LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_S181)
    LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_S181)
  ,
  SUBPARTITION P2011JUL_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011JUL_SMAX
    LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_SMAX)
    LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_SMAX)
    LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_SMAX)
    LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_SMAX)
    LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_SMAX)
),
PARTITION "P2011AUG" VALUES LESS THAN (TO_DATE('2011-09-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))( 
  SUBPARTITION P2011AUG_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011AUG_S181
    LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_S181)
    LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_S181)
    LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_S181)
    LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_S181)
    LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_S181)
  ,
  SUBPARTITION P2011AUG_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011AUG_SMAX
    LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_SMAX)
    LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_SMAX)
    LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_SMAX)
    LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_SMAX)
    LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_SMAX)
PARTITION "P2011SEP" VALUES LESS THAN (TO_DATE('2011-10-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')) ( 
  SUBPARTITION P2011SEP_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011SEP_S181 
  - LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_S181) 
  - LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_S181) 
  - LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_S181) 
  - LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_S181) 
  - LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_S181) 
  , 
  SUBPARTITION P2011SEP_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011SEP_SMAX 
  - LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_SMAX) 
  - LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_SMAX) 
  - LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_SMAX) 
  - LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_SMAX) 
  - LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_SMAX) 
), 
PARTITION "P2011OCT" VALUES LESS THAN (TO_DATE('2011-11-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')) ( 
  SUBPARTITION P2011OCT_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011OCT_S181 
  - LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_S181) 
  - LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_S181) 
  - LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_S181) 
  - LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_S181) 
  - LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_S181) 
  , 
  SUBPARTITION P2011OCT_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011OCT_SMAX 
  - LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_SMAX) 
  - LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_SMAX) 
  - LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_SMAX) 
  - LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_SMAX) 
  - LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_SMAX) 
), 
PARTITION "P2011NOV" VALUES LESS THAN (TO_DATE('2011-12-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')) ( 
  SUBPARTITION P2011NOV_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011NOV_S181 
  - LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_S181) 
  - LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_S181) 
  - LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_S181) 
  - LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_S181) 
  - LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_S181) 
  )

LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_S181)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_S181)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_S181)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_S181)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_S181)
SUBPARTITION P2011NOV_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011NOV_SMAX
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_SMAX)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_SMAX)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_SMAX)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_SMAX)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_SMAX)
PARTITION "P2011DEC" VALUES LESS THAN (TO_DATE('2012-01-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
SUBPARTITION P2011DEC_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011DEC_S181
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_S181)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_S181)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_S181)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_S181)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_S181)
SUBPARTITION P2011DEC_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011DEC_SMAX
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_SMAX)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_SMAX)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_SMAX)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_SMAX)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_SMAX)
SUBPARTITION "PMAX" VALUES LESS THAN (MAXVALUE)
SUBPARTITION PMAX_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_PMAX_S181
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_S181)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_S181)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_S181)
INDEX

CREATE BIGFILE TABLESPACE CM_D1T304_IND DATAFILE '+DATA' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;

CREATE UNIQUE INDEX D1T304P0 ON D1_INIT_MSRMT_DATA(INIT_MSRMT_DATA_ID) TABLESPACE CM_D1T304_IND
GLOBAL PARTITION BY RANGE (INIT_MSRMT_DATA_ID)
( PARTITION P1 VALUES LESS THAN ( '12499999999999' ),
PARTITION P2 VALUES LESS THAN ( '24999999999999' ),
PARTITION P3 VALUES LESS THAN ( '37499999999999' ),
PARTITION P4 VALUES LESS THAN ( '49999999999999' ),
PARTITION P5 VALUES LESS THAN ( '62499999999999' ),
PARTITION P6 VALUES LESS THAN ( '74999999999999' ),
PARTITION P7 VALUES LESS THAN ( '87499999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
);

ALTER TABLE D1_INIT_MSRMT_DATA ADD CONSTRAINT D1T304P0 PRIMARY KEY(INIT_MSRMT_DATA_ID) USING INDEX;

CREATE INDEX D1T304S1 ON D1_INIT_MSRMT_DATA (MEASR_COMP_ID, BO_STATUS_CD, BUS_OBJ_CD, D1_TO_DTTM, D1_FROM_DTTM) TABLESPACE CM_D1T304_IND
GLOBAL PARTITION BY RANGE (MEASR_COMP_ID)
( PARTITION P1 VALUES LESS THAN ( '12499999999999' ),
PARTITION P2 VALUES LESS THAN ( '24999999999999' ),
PARTITION P3 VALUES LESS THAN ( '37499999999999' ),
PARTITION P4 VALUES LESS THAN ( '49999999999999' ),
PARTITION P5 VALUES LESS THAN ( '62499999999999' ),
PARTITION P6 VALUES LESS THAN ( '74999999999999' ),
PARTITION P7 VALUES LESS THAN ( '87499999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
)
COMPRESSION ADVANCED LOW;

CREATE UNIQUE INDEX D1T304S3 ON
D1_INIT_MSRMT_DATA(IMD_EXT_ID, INIT_MSRMT_DATA_ID)
GLOBAL PARTITION BY HASH(IMD_EXT_ID)
CREATE TABLE D1_INIT_MSRMT_DATA_CHAR
(
  INIT_MSRMT_DATA_ID CHAR(14) NOT NULL ENABLE,
  CHAR_TYPE_CD CHAR(8) NOT NULL ENABLE,
  SEQ_NUM NUMBER(3,0) NOT NULL ENABLE,
  CHAR_VAL CHAR(16) DEFAULT ' ' NOT NULL ENABLE,
  ADHOC_CHAR_VAL VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK1 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK2 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK3 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK4 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK5 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  SRCH_CHAR_VAL VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
  VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
  LAST_UPDATE_DTTM DATE,
  CONSTRAINT D1_INIT_MSRMT_DATA_CHAR_FK FOREIGN KEY(INIT_MSRMT_DATA_ID) REFERENCES D1_INIT_MSRMT_DATA ON DELETE CASCADE)
PARTITION BY REFERENCE (D1_INIT_MSRMT_DATA_CHAR_FK)
ENABLE ROW MOVEMENT;

INDEX

CREATE UNIQUE INDEX D1T305P0 ON
D1_INIT_MSRMT_DATA_CHAR(INIT_MSRMT_DATA_ID, CHAR_TYPE_CD, SEQ_NUM)
TABLESPACE CM_D1T304_IND
GLOBAL PARTITION BY RANGE(INIT_MSRMT_DATA_ID)
(
  PARTITION P1 VALUES LESS THAN ('12499999999999'),
  PARTITION P2 VALUES LESS THAN ('24999999999999'),
  PARTITION P3 VALUES LESS THAN ('37499999999999'),
  PARTITION P4 VALUES LESS THAN ('49999999999999'),
  PARTITION P5 VALUES LESS THAN ('62499999999999'),
  PARTITION P6 VALUES LESS THAN ('74999999999999'),
  PARTITION P7 VALUES LESS THAN ('87499999999999'),
  PARTITION P8 VALUES LESS THAN (MAXVALUE)
) COMPRESS ADVANCED LOW;
ALTER TABLE D1_INIT_MSRMT_DATA_CHAR ADD CONSTRAINT D1T305P0 PRIMARY KEY (INIT_MSRMT_DATA_ID, CHAR_TYPE_CD, SEQ_NUM) USING INDEX;

CREATE INDEX D1T305S1 ON D1_INIT_MSRMT_DATA_CHAR(SRCH_CHAR_VAL) GLOBAL PARTITION BY HASH(SRCH_CHAR_VAL)

PARTITION P1 TABLESPACE CM_D1T304_IND,
PARTITION P2 TABLESPACE CM_D1T304_IND,
PARTITION P3 TABLESPACE CM_D1T304_IND,
PARTITION P4 TABLESPACE CM_D1T304_IND,
PARTITION P5 TABLESPACE CM_D1T304_IND,
PARTITION P6 TABLESPACE CM_D1T304_IND,
PARTITION P7 TABLESPACE CM_D1T304_IND,
PARTITION P8 TABLESPACE CM_D1T304_IND
);

CREATE TABLE D1_INIT_MSRMT_DATA_LOG

(INIT_MSRMT_DATA_ID CHAR(14) NOT NULL ENABLE,
SEQNO NUMBER(5,0) NOT NULL ENABLE,
BO_STATUS_CD CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
BO_STATUS_REASON_CD VARCHAR2(30) DEFAULT ' ' NOT NULL ENABLE,
CHAR_TYPE_CD CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL CHAR(16) DEFAULT ' ' NOT NULL ENABLE,
ADHOC_CHAR_VAL VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK1 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK2 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK3 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK4 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK5 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
DESCRLONG VARCHAR2(4000) DEFAULT ' ' NOT NULL ENABLE,
LOG_DTTM DATE NOT NULL ENABLE,
LOG_ENTRY_TYPE_FLG CHAR(4) DEFAULT ' ' 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) DEFAULT ' ' NOT NULL ENABLE,
VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
LAST_UPDATE_DTTM DATE,
CONSTRAINT D1_INIT_MSRMT_DATA_LOG_FK FOREIGN KEY(INIT_MSRMT_DATA_ID) REFERENCES D1_INIT_MSRMT_DATA ON DELETE CASCADE)
PARTITION BY REFERENCE (D1_INIT_MSRMT_DATA_LOG_FK)
ENABLE ROW MOVEMENT;

INDEX

CREATE UNIQUE INDEX D1T306P0 ON D1_INIT_MSRMT_DATA_LOG
(INIT_MSRMT_DATA_ID, SEQNO) TABLESPACE CM_D1T304_IND
GLOBAL PARTITION BY RANGE(INIT_MSRMT_DATA_ID)

PARTITION P1 VALUES LESS THAN ('12499999999999'),
PARTITION P2 VALUES LESS THAN ('24999999999999'),
PARTITION P3 VALUES LESS THAN ('37499999999999'),
PARTITION P4 VALUES LESS THAN ('49999999999999'),
PARTITION P5 VALUES LESS THAN ('62499999999999'),
Child Table: D1_INIT_MSRMT_DATA_LOG_PARM

CREATE TABLE D1_INIT_MSRMT_DATA_LOG_PARM
(
    INIT_MSRMT_DATA_ID CHAR(14) NOT NULL ENABLE,
    SEQNO              NUMBER(5,0) NOT NULL ENABLE,
    PARM_SEQ           NUMBER(3,0) NOT NULL ENABLE,
    MSG_PARM_VAL       VARCHAR2(2000) DEFAULT ' ' NOT NULL ENABLE,
    MSG_PARM_TYP_FLG   CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
    VERSION            NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    LAST_UPDATE_DTTM DATE,
    CONSTRAINT D1_INIT_MSRMT_DATA_LOG_PARM_FK FOREIGN KEY(INIT_MSRMT_DATA_ID) REFERENCES D1_INIT_MSRMT_DATA ON DELETE CASCADE
) PARTITION BY REFERENCE (D1_INIT_MSRMT_DATA_LOG_PARM_FK)
ENABLE ROW MOVEMENT;

INDEX

CREATE UNIQUE INDEX D1T307P0 ON D1_INIT_MSRMT_DATA_LOG_PARM
(INIT_MSRMT_DATA_ID, SEQNO, PARM_SEQ)
TABLESPACE CM_D1T304_IND
GLOBAL PARTITION BY RANGE(INIT_MSRMT_DATA_ID)
(
    PARTITION P1 VALUES LESS THAN ('12499999999999'),
    PARTITION P2 VALUES LESS THAN ('24999999999999'),
    PARTITION P3 VALUES LESS THAN ('37499999999999'),
    PARTITION P4 VALUES LESS THAN ('49999999999999'),
    PARTITION P5 VALUES LESS THAN ('62499999999999'),
    PARTITION P6 VALUES LESS THAN ('74999999999999'),
    PARTITION P7 VALUES LESS THAN ('87499999999999'),
    PARTITION P8 VALUES LESS THAN (MAXVALUE)
) COMRESS ADVANCED LOW;

ALTER TABLE D1_INIT_MSRMT_DATA_LOG_ADD CONSTRAINT D1T307P0 PRIMARY KEY
(INIT_MSRMT_DATA_ID, SEQNO, PARM_SEQ) USING INDEX;

Child Table: D1_INIT_MSRMT_DATA_K

CREATE BIGFILE TABLESPACE CM_D1T314_IND DATAFILE '+DATA' SIZE 50M
AUTOEXTEND ON MAXSIZE UNLIMITED;

CREATE TABLE D1_INIT_MSRMT_DATA_K
(  INIT_MSRMT_DATA_ID CHAR(14) NOT NULL ENABLE,
  ENV_ID NUMBER(6,0) NOT NULL ENABLE,
  CONSTRAINT D1T314P0 PRIMARY KEY (INIT_MSRMT_DATA_ID, ENV_ID) ENABLE  
)

ORGANIZATION INDEX
  Partition by range(INIT_MSRMT_DATA_ID)
  
  {  
  PARTITION P1 VALUES LESS THAN ( '12499999999999' ),
  PARTITION P2 VALUES LESS THAN ( '24999999999999' ),
  PARTITION P3 VALUES LESS THAN ( '37499999999999' ),
  PARTITION P4 VALUES LESS THAN ( '49999999999999' ),
  PARTITION P5 VALUES LESS THAN ( '62499999999999' ),
  PARTITION P6 VALUES LESS THAN ( '74999999999999' ),
  PARTITION P7 VALUES LESS THAN ( '87499999999999' ),
  PARTITION P8 VALUES LESS THAN ( MAXVALUE )
  }

TABLESPACE CM_D1T314_IND;
This section provides additional details related to supporting ILM in an existing installation. It includes 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:

```sql
ALTER TABLE CI_TD_ENTRY RENAME TO ILM_TD_ENTRY;
ALTER INDEX XT039P0 RENAME TO ILM_XT039P0;
```

2. Generate DDL for the secondary index.

```sql
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.

```sql
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.

```sql
CREATE TABLE CI_TD_ENTRY (  
TD_ENTRY_ID CHAR(14) NOT NULL ENABLE,  
BATCH_CD CHAR(8) DEFAULT ' ' NOT NULL ENABLE,  
BATCH_NBR NUMBER(10,0) DEFAULT 0 NOT NULL ENABLE,  
MESSAGE_CAT_NBR NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,  
MESSAGE_NBR NUMBER(5,0) DEFAULT 0 NOT NULL ENABLE,  
```

```sql
);  
```

Sample SQL For Enabling ILM in C2M for MDM (Existing Installation)
ASSIGNED_TO CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
TD_TYPE_CD CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
ROLE_ID CHAR(10) DEFAULT ' ' NOT NULL ENABLE,
ENTRY_STATUS_FLG CHAR(2) DEFAULT ' ' NOT NULL ENABLE,
VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
CRE_DTTM DATE,
ASSIGNED DTTM DATE,
COMPLETE_DTTM DATE,
COMPLETE_USER_ID CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
COMMENTS VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
ASSIGNED_USER_ID CHAR(8) DEFAULT ' ' NOT NULL ENABLE,
TD_PRIORITY_FLG CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
ILM_DT DATE,
ILM_ARCH_SW CHAR(1)
) NOLOGGING PARALLEL
ENABLE ROW MOVEMENT
PARTITION BY RANGE (ILM_DT)
SUBPARTITION BY RANGE (TD_ENTRY_ID) SUBPARTITION TEMPLATE
|
| SUBPARTITION S01 VALUES LESS THAN ( '12499999999999' ),
| SUBPARTITION S02 VALUES LESS THAN ( '24999999999999' ),
| SUBPARTITION S03 VALUES LESS THAN ( '37499999999999' ),
| SUBPARTITION S04 VALUES LESS THAN ( '49999999999999' ),
| SUBPARTITION S05 VALUES LESS THAN ( '62499999999999' ),
| SUBPARTITION S06 VALUES LESS THAN ( '74999999999999' ),
| SUBPARTITION S07 VALUES LESS THAN ( '87499999999999' ),
| SUBPARTITION SMAX VALUES LESS THAN ( MAXVALUE )
|
| PARTITION "P2017JAN" VALUES LESS THAN (TO_DATE('2017-02-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017JAN,
PARTITION "P2017FEB" VALUES LESS THAN (TO_DATE('2017-03-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017FEB,
PARTITION "P2017MAR" VALUES LESS THAN (TO_DATE('2017-04-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017MAR,
PARTITION "P2017APR" VALUES LESS THAN (TO_DATE('2017-05-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017APR,
PARTITION "P2017MAY" VALUES LESS THAN (TO_DATE('2017-06-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017MAY,
PARTITION "P2017JUN" VALUES LESS THAN (TO_DATE('2017-07-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017JUN,
PARTITION "P2017JUL" VALUES LESS THAN (TO_DATE('2017-08-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017JUL,
PARTITION "P2017AUG" VALUES LESS THAN (TO_DATE('2017-09-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017AUG,
PARTITION "P2017SEP" VALUES LESS THAN (TO_DATE('2017-10-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017SEP,
PARTITION "P2017OCT" VALUES LESS THAN (TO_DATE('2017-11-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017OCT,
PARTITION "P2017NOV" VALUES LESS THAN (TO_DATE('2017-12-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017NOV,
PARTITION "P2017DEC" VALUES LESS THAN (TO_DATE('2018-01-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace CM_XT039_P2017DEC,
PARTITION "PMAX" VALUES LESS THAN (MAXVALUE)
tablespace CM_XT039
) as select /* PARALLEL */
) as select /* PARALLEL */
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,
5. Enable logging option for table CI_TD_ENTRY.
   
   ALTER TABLE CI_TD_ENTRY NOPARALLEL LOGGING;

6. Create Primary Index for Parent table CI_TD_ENTRY.
   
   CREATE BIGFILE TABLESPACE CM_XT039_IND DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;

   CREATE UNIQUE INDEX XT039P0 ON CI_TD_ENTRY NOLOGGING PARALLEL (TD_ENTRY_ID)

   PARTITION P1 VALUES LESS THAN (1249999999999999),
   PARTITION P2 VALUES LESS THAN (2499999999999999),
   PARTITION P3 VALUES LESS THAN (3749999999999999),
   PARTITION P4 VALUES LESS THAN (4999999999999999),
   PARTITION P5 VALUES LESS THAN (6249999999999999),
   PARTITION P6 VALUES LESS THAN (7499999999999999),
   PARTITION P7 VALUES LESS THAN (8749999999999999),
   PARTITION P8 VALUES LESS THAN (MAXVALUE)

   TABLESPACE CM_XT039_IND

   ALTER INDEX XT039P0 LOGGING NOPARALLEL;

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 CM_ILM_XT039S8 ON CI_TD_ENTRY (ILM_DT, ILM_ARCH_SW, TD_ENTRY_ID)

   CREATE UNIQUE INDEX XT039S2 ON CI_TD_ENTRY (ASSIGNED_TO, TD_ENTRY_ID)

   CREATE UNIQUE 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 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 all child Tables, Primary Key, Primary Indexes and Secondary Indexes as shown below.

Repeat the following steps for all child tables.

Create Child Table CI_TD_DRLKEY
   
   CREATE TABLE CI_TD_DRLKEY
   (TD_ENTRY_ID NOT NULL ENABLE,
    SEQ_NUM NOT NULL ENABLE,
KEY_VALUE DEFAULT ' ' NOT NULL ENABLE,
VERSION DEFAULT 1 NOT NULL ENABLE,
CONSTRAINT CI_TD_DRLKEY_FK FOREIGN KEY(TD_ENTRY_ID) REFERENCES CI_TD_ENTRY ON DELETE
CASCADE
)
PARTITION BY REFERENCE (CI_TD_DRLKEY_FK)
ENABLE ROW MOVEMENT
AS SELECT /*+ PARALLEL */ * FROM ILM_CI_TD_DRLKEY;

Create Index

CREATE UNIQUE INDEX XT037P0 ON CI_TD_DRLKEY ( TD_ENTRY_ID, SEQ_NUM ) TABLESPACE
CM_XT039_IND NOLOGGING PARALLEL
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 ( '749999999999' ),
PARTITION P7 VALUES LESS THAN ( '874999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
)
COMPRESS ADVANCED LOW;
ALTER INDEX XT037P0 LOGGING NOPARALLEL;
ALTER TABLE CI_TD_DRLKEY ADD CONSTRAINT XT037P0 PRIMARY KEY(TD_ENTRY_ID, SEQ_NUM) USING
INDEX;
CREATE INDEX XT037S1 ON CI_TD_DRLKEY ( KEY_VALUE, TD_ENTRY_ID ) TABLESPACE CM_XT039_IND
COMPRESS ADVANCED LOW;
This section provides additional details including the sample syntax for each step using the Initial Measurement Data maintenance object as an example. Other maintenance object's implementations can follow a similar pattern.

1. Rename existing D1_INIT_MSRMT_DATA tables and primary key indexes and constraints as a backup. It is suggested to use an ILM_prefix. The following are sample statements:

   ALTER TABLE D1_INIT_MSRMT_DATA RENAME TO ILM_D1_INIT_MSRMT_DATA;

   ALTER TABLE D1_INIT_MSRMT_DATA RENAME CONSTRAINT D1T304P0 TO ILM_D1T304P0;

   ALTER INDEX D1T304P0 RENAME TO ILM_D1T304P0;

   ALTER TABLE D1_INIT_MSRMT_DATA_CHAR RENAME TO ILM_D1_INIT_MSRMT_DATA_CHAR;

   ALTER TABLE D1_INIT_MSRMT_DATA_CHAR RENAME CONSTRAINT D1T305P0 TO ILM_D1T305P0;

   ALTER INDEX D1T305P0 RENAME TO ILM_D1T305P0;

   ALTER TABLE D1_INIT_MSRMT_DATA_LOG RENAME TO ILM_D1_INIT_MSRMT_DATA_LOG;

   ALTER TABLE D1_INIT_MSRMT_DATA_LOG RENAME CONSTRAINT D1T306P0 TO ILM_D1T306P0;

   ALTER INDEX D1T306P0 RENAME TO ILM_D1T306P0;

   ALTER TABLE D1_INIT_MSRMT_DATA_LOG_PARM RENAME TO ILM_D1_INIT_MSRMT_DATA_LOG_PARM;

   ALTER TABLE D1_INIT_MSRMT_DATA_LOG_PARM RENAME CONSTRAINT D1T307P0 TO ILM_D1T307P0;

   ALTER INDEX D1T307P0 RENAME TO ILM_D1T307P0;

   ALTER TABLE D1_INIT_MSRMT_DATA_K RENAME TO ILM_D1_INIT_MSRMT_DATA_K;
ALTER TABLE D1_INIT_MSRMT_DATA_K RENAME CONSTRAINT D1T314P0 TO ILM_D1T314P0;

ALTER INDEX D1T314P0 RENAME TO ILM_D1T314P0;

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','D1T304S1','CISADM') from dual;
   select dbms_metadata.get_ddl('INDEX','D1T304S3','CISADM') from dual;
   spool off;

3. Drop secondary indexes.
   DROP INDEX CISADM.D1T304S1;
   DROP INDEX CISADM.D1T304S3;

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. Use the CTAS queries listed in Chapter 5 to create temporary tables for ACTIVITY, DEVICE EVENT, and INITIAL MEASUREMENT DATA and use the following statements to create the partitioned tables.

**Activity**

```sql
CREATE TABLE D1_ACTIVITY (  
    D1_ACTIVITY_ID      NOT NULL,
    BUS_OBJ_CD          NOT NULL,
    BO_STATUS_CD        NOT NULL,
    ACTIVITY_TYPE_CD    NOT NULL,
    START_DTTM          NOT NULL,
    END_DTTM,
    CRE_DTTM            NOT NULL,
    STATUS_UPD_DTTM     NOT NULL,
    BO_STATUS_REASON_CD NOT NULL,
    VERSION             NOT NULL,
    EFF_DTTM,
    BO_DATA_AREA,
    FIELD_TASK_TYPE,
    CANCEL_REASON,
    ILM_DT,
    ILM_ARCH_SW,
    RETENTION_PERIOD    NOT NULL  )  
AS  
SELECT  
A.D1_ACTIVITY_ID,  
A.BUS_OBJ_CD,  
A.BO_STATUS_CD,  
A.ACTIVITY_TYPE_CD,  
A.START_DTTM,  
A.END_DTTM,  
A.CRE_DTTM,  
A.STATUS_UPD_DTTM,
```
A.BO_STATUS_REASON_CD,
A.VERSION,
A.EFF_DTTM,
A.BO_DATA_AREA,
A.FIELD_TASK_TYPE,
A.CANCEL_REASON,
A.CRE_DTTM as ILM_DT,
'N' as ILM_ARCH_SW,
CAST(COALESCE((SELECT B.RETPERIOD
FROM ILM_ACTIVITY_RETENTION_TMP B
WHERE B.ACTIVITY_TYPE_CD = A.ACTIVITY_TYPE_CD)
,CAST((select maint_obj_opt_val
from ci_md_mo_opt mmouni
where maint_obj_cd = 'D1-ACTIVITY'
and maint_obj_opt_flg = 'FLRP'
and seq_num =
(select max(seq_num)
from ci_md_mo_opt mmo
where maint_obj_cd = 'D1-ACTIVITY'
and maint_obj_opt_flg = 'FLRP')
)) as NUMBER(5)),
CAST((select extractvalue( xmlparse(content fw_mcfg.mst_config_data)
,'generalMasterConfiguration/defaultRetentionPeriod')
from f1_mst_config fw_mcfg
where fw_mcfg.bus_obj_cd = 'F1-ILMMSConfig') as NUMBER(5),
99999) as NUMBER(5)) as RETENTION_PERIOD
FROM ILM_D1_ACTIVITY A
/

Device Event
CREATE TABLE D1_DVC_EVT(
DVC_EVT_ID          NOT NULL,
DVC_EVT_TYPE_CD,    NOT NULL,
BUS_OBJ_CD          NOT NULL,
EXT_EVT_NAME_FLG,   NOT NULL,
D1_SPR_CD,          NOT NULL,
BO_STATUS_CD        NOT NULL,
STATUS_UPD_DTTM,    NOT NULL,
BO_STATUS_REASON_CD NOT NULL,
DVC_EVT_DTTM,       NOT NULL,
CRE_DTTM,           NOT NULL,
VERSION,            NOT NULL,
DVC_EVT_END_DTTM,   NOT NULL,
BO_DATA_AREA,       NOT NULL,
D1_DEVICE_ID,       NOT NULL,
ILM_DT              NOT NULL,
ILM_ARCH_SW,        NOT NULL,
RETENTION_PERIOD    NOT NULL)
AS
SELECT A.DVC_EVT_ID,
A.DVC_EVT_TYPE_CD,
A.BUS_OBJ_CD,
A.EXT_EVT_NAME_FLG,
A.D1_SPR_CD,
A.BO_STATUS_CD,
A.STATUS_UPD_DTTM,
A.BO_STATUS_REASON_CD,
A.DVC_EVT_DTTM,
A.CRE_DTTM,
A.VERSIO
A.DVC_EVT_END_DTTM,
A.BO_DATA_AREA,
A.D1_DEVICE_ID,
A.CRE_DTTM as ILM_DT,
'N' as ILM_ARCH_SW,
CAST(COALESCE((SELECT B.RETPERIOD
FROM ILM_DVC_EVT_RETENTION_TMP B
WHERE B.DVC_EVT_TYPE_CD = A.DVC_EVT_TYPE_CD)
,CAST((select maint_obj_opt_val
from ci_md_mo_opt mmouni
where maint_obj_cd = 'D1-DVC_EVENT' and maint_obj_opt_flg = 'FLRP'
and seq_num = (select max(seq_num)
from ci_md_mo_opt mmo
where maint_obj_cd = 'D1-DVC_EVENT' and maint_obj_opt_flg = 'FLRP')
,CAST((select extractvalue( xmlparse(content fw_mcfg.mst_config_data)
,'generalMasterConfiguration/defaultRetentionPeriod')
from f1_mst_config fw_mcfg
where fw_mcfg.bus_obj_cd = 'F1-ILMMSConfig') as NUMBER(5))
,99999) as NUMBER(5)) as RETENTION_PERIOD
FROM ILM_D1_DVC_EVT A
/

Initial Measurement Data

CREATE TABLE ILM_IMD_RETENTION_TMP
AS
select mct.measr_comp_type_cd
/*retrieve the retention period for MC Types in this order of precedence:
1. The UOM based retention period from the MDM master configuration
2. The interval IMD retention period from the MDM master configuration
3. The MO level retention period from the MO options
4. The installation level retention period from the FW master configuration*/
CAST(coalesce( (select 'D1IN' interval_scalar_flg,
extractvalue(value(p),'uomRetentionPeriodList/uom') D1_UOM_CD,
extractvalue(value(p),'uomRetentionPeriodList/retentionPeriod') retPeriod
from f1_mst_config mdm_mcfg,
,table(xmlsequence(extract(xmlparse(content
mdm_mcfg.mst_config_data),
'imdRetentionPeriod/intervalImdRetentionPeriods/uomRetentionPeriods/
imdRetentionPeriodList')) p
where mdm_mcfg.bus_obj_cd = 'D1-ILMMSConfig')
union
select 'D1SC' INTERVAL_SCALAR_FLG,
extractvalue(value(p),'uomRetentionPeriodList/uom') D1_UOM_CD,
extractvalue(value(p),'uomRetentionPeriodList/retentionPeriod') retPeriod
from f1_mst_config mdm_mcfg,
,table(xmlsequence(extract(xmlparse(content
mdm_mcfg.mst_config_data),
'imdRetentionPeriod/scalarImdRetentionPeriods/uomRetentionPeriods/
imdRetentionPeriodList')) p
where mdm_mcfg.bus_obj_cd = 'D1-ILMMSConfig') uomMap
where uomMap.interval_scalar_flg = mct.interval_scalar_flg
and trim(mctvi.d1_uom_cd) = trim(uomMap.d1_uom_cd) -- UOM
and decode(mct.interval_scalar_flg, 'D1IN',
extractvalue( xmlparse(content mdm_mcfg.mst_config_data),
'imdRetentionPeriod/intervalImdRetentionPeriods/
intervalRetentionPeriod') -- interval IMD
, extractvalue( xmlparse(content mdm_mcfg.mst_config_data),
'imdRetentionPeriod/scalarImdRetentionPeriods/scalarRetentionPeriod')
-- scalar IMD
, (select main_obj_opt_val
from ci_md_mo_opt mmo
where main_obj_cd = 'D1-IMD'
and main_obj_opt_flg = 'FLRP'
and seq_num = (select max(seq_num)
from ci_md_mo_opt mmo
where main_obj_cd = 'D1-IMD'
and main_obj_opt_flg = 'FLRP') -- IMD
, extractvalue( xmlparse(content fw_mcfg.mst_config_data),
'generalMasterConfiguration/defaultRetentionPeriod') -- Install
) as NUMBER(5)) retPeriod
from d1_measr_comp_type mct
, d1_mc_type_value_identifier mctvi
, f1_mst_config fw_mcfg
, f1_mst_config mdm_mcfg
where mct measr_comp_type_cd = mctvi measr_comp_type_cd
and mctvi.value_id_type_flg = 'D1MS'
and fw_mcfg.bus_obj_cd = 'F1-ILMMSConfig'
and mdm_mcfg.bus_obj_cd = 'D1-ILMMSConfig'
order by 1;

CREATE BIGFILE TABLESPACE CM_D1T304_P2011JAN_S181 DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011JAN_SMAX DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011FEB_S181 DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011FEB_SMAX DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011MAR_S181 DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011MAR_SMAX DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011APR_S181 DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011APR_SMAX DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011MAY_S181 DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011MAY_SMAX DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011JUN_S181 DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011JUN_SMAX DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011JUL_S181 DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011JUL_SMAX DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011AUG_S181 DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011AUG_SMAX DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011SEP_S181 DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011SEP_SMAX DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011OCT_S181 DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011OCT_SMAX DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011NOV_S181 DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011NOV_SMAX DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011DEC_S181 DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011DEC_SMAX DATAFILE '+DATADG'
SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011MAY_SMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011JUN_S181 DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011JUN_SMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011JUL_S181 DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011JUL_SMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011AUG_S181 DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011AUG_SMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011SEP_S181 DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011SEP_SMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011OCT_S181 DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011OCT_SMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011NOV_S181 DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011NOV_SMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011DEC_S181 DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011DEC_SMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_PMAX_S181 DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE BIGFILE TABLESPACE CM_D1T304_PMAX_SMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;

CREATE TABLE D1_INIT_MSRMT_DATA
(
  INIT_MSRMT_DATA_ID   NOT NULL,
  MEASR_COMP_ID        NOT NULL,
  D1_FROM_DTTM,
  D1_TO_DTTM,
  DATA_SRC_FLG         NOT NULL,
  TIME_ZONE_CD         NOT NULL,
  BUS_OBJ_CD           NOT NULL,
BO_STATUS_CD         NOT NULL,
BO_STATUS_REASON_CD  NOT NULL,
IMD_BO_DATA_AREA,
STATUS_UPD_DTTM      NOT NULL,
CRE_DTTM             NOT NULL,
VERSION              NOT NULL,
IMD_EXT_ID,
PREVEE_BO_DATA_AREA,
POSTVEE_BO_DATA_AREA,
TRACE_BO_DATA_AREA,
RAW_BO_DATA_AREA,
LAST_UPDATE_DTTM,
ILM_DT,
ILM_ARCH_SW,
RETENTION_PERIOD     NOT NULL
)

nologging  parallel (degree 10)
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW
COMPRESS MEDIUM CACHE)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW
COMPRESS MEDIUM CACHE)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW
COMPRESS MEDIUM CACHE)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW
COMPRESS MEDIUM CACHE)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW
COMPRESS MEDIUM CACHE)
PARTITION BY RANGE (ILM_DT) SUBPARTITION BY RANGE (RETENTION_PERIOD)
{
  PARTITION "P2011JAN" VALUES LESS THAN (TO_DATE('2011-02-01 00:00:01',
    LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_S181)
    LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_S181)
    LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_S181)
    LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_S181)
    LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_S181)

  ,
  SUBPARTITION P2011JAN_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011JAN_SMAX
    LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_SMAX)
    LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_SMAX)
    LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_SMAX)
    LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_SMAX)
    LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN_SMAX)
  ),
  PARTITION "P2011FEB" VALUES LESS THAN (TO_DATE('2011-03-01 00:00:01',
    LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)
    LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)
    LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)
    LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)
    LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181

  ,
  SUBPARTITION P2011FEB_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011FEB_SMAX
    LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_SMAX)
    LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_SMAX)
    LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_SMAX)
    LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_SMAX)
    LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_SMAX

  )

);
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_S181)
SUBPARTITION P2011FEB_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011FEB_SMAX
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_SMAX)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_SMAX)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_SMAX)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_SMAX)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011FEB_SMAX)
PARTITION "P2011MAR" VALUES LESS THAN (TO_DATE('2011-04-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))(SUBPARTITION P2011MAR_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011MAR_S181
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_S181)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_S181)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_S181)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_S181)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_S181)
SUBPARTITION P2011MAR_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011MAR_SMAX
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_SMAX)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_SMAX)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_SMAX)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_SMAX)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR_SMAX)
PARTITION "P2011APR" VALUES LESS THAN (TO_DATE('2011-05-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))(SUBPARTITION P2011APR_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011APR_S181
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_S181)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_S181)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_S181)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_S181)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_S181)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_S181)
- LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_S181)
),
SUBPARTITION P2011APR_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011APR_SMAX
- LOB (PREV_EE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_SMAX)
- LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_SMAX)
- LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_SMAX)
- LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_SMAX)
- LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011APR_SMAX)
),
PARTITION "P2011MAY" VALUES LESS THAN (TO_DATE('2011-06-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))(
SUBPARTITION P2011MAY_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011MAY_S181
- LOB (PREV_EE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_S181)
- LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_S181)
- LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_S181)
- LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_S181)
- LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_S181)
),
SUBPARTITION P2011MAY_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011MAY_SMAX
- LOB (PREV_EE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_SMAX)
- LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_SMAX)
- LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_SMAX)
- LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_SMAX)
- LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY_SMAX)
),
PARTITION "P2011JUN" VALUES LESS THAN (TO_DATE('2011-07-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))(
SUBPARTITION P2011JUN_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011JUN_S181
- LOB (PREV_EE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_S181)
- LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_S181)
- LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_S181)
- LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_S181)
- LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_S181)
),
SUBPARTITION P2011JUN_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011JUN_SMAX
- LOB (PREV_EE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
- LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
- LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
- LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
- LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
SUBPARTITION P2011JUN_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011JUN_SMAX
  LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
  LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
  LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
  LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
  LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUN_SMAX)
),
PARTITION "P2011JUL" VALUES LESS THAN (TO_DATE('2011-08-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  SUBPARTITION P2011JUL_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011JUL_S181
  LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_S181)
  LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_S181)
  LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_S181)
  LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_S181)
  LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_S181)
  SUBPARTITION P2011JUL_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011JUL_SMAX
  LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_SMAX)
  LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_SMAX)
  LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_SMAX)
  LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_SMAX)
  LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL_SMAX)
),
PARTITION "P2011AUG" VALUES LESS THAN (TO_DATE('2011-09-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  SUBPARTITION P2011AUG_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011AUG_S181
  LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_S181)
  LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_S181)
  LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_S181)
  LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_S181)
  LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_S181)
  SUBPARTITION P2011AUG_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011AUG_SMAX
  LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_SMAX)
  LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_SMAX)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_SMAX)
  LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_SMAX)
  LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011AUG_SMAX)
),
PARTITION "P2011SEP" VALUES LESS THAN (TO_DATE('2011-10-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))(
  SUBPARTITION P2011SEP_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011SEP_S181
  LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_S181)
  LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_S181)
  LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_S181)
  LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_S181)
  LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_S181)
  SUBPARTITION P2011SEP_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011SEP_SMAX
  LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_SMAX)
  LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_SMAX)
  LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_SMAX)
  LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_SMAX)
  LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP_SMAX)
),
PARTITION "P2011OCT" VALUES LESS THAN (TO_DATE('2011-11-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))(
  SUBPARTITION P2011OCT_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011OCT_S181
  LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_S181)
  LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_S181)
  LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_S181)
  LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_S181)
  LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_S181)
  SUBPARTITION P2011OCT_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011OCT_SMAX
  LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_SMAX)
  LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_SMAX)
  LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_SMAX)
  LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_SMAX)
  LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011OCT_SMAX)
Sample SQL for ILM in C2M with Sub Retention (Existing Installation)

),
PARTITION "P2011NOV" VALUES LESS THAN (TO_DATE('2011-12-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))( SUBPARTITION P2011NOV_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011NOV_S181
  _LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_S181)
  _LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_S181)
  _LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_S181)
  _LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_S181)
  _LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_S181)
  _SUBPARTITION P2011NOV_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011NOV_SMAX
  _LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_SMAX)
  _LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_SMAX)
  _LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_SMAX)
  _LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_SMAX)
  _LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV_SMAX)
),
PARTITION "P2011DEC" VALUES LESS THAN (TO_DATE('2012-01-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))( SUBPARTITION P2011DEC_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_P2011DEC_S181
  _LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_S181)
  _LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_S181)
  _LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_S181)
  _LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_S181)
  _LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_S181)
  _SUBPARTITION P2011DEC_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_P2011DEC_SMAX
  _LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_SMAX)
  _LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_SMAX)
  _LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_SMAX)
  _LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_SMAX)
  _LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011DEC_SMAX)
),
PARTITION "PMAX" VALUES LESS THAN (MAXVALUE)( SUBPARTITION PMAX_S181 VALUES LESS THAN (181) TABLESPACE CM_D1T304_PMAX_S181
  _LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_S181)
  _LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_S181)
  _LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_S181)
  _LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_S181)
  _LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_S181)
)
Sample SQL for ILM in C2M with Sub Retention (Existing Installation)

LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_S181)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_S181)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_S181)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_S181)
,
SUBPARTITION PMAX_SMAX VALUES LESS THAN (MAXVALUE) TABLESPACE CM_D1T304_PMAX_SMAX
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_SMAX)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_SMAX)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_SMAX)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_SMAX)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX_SMAX)
)) ENABLE ROW MOVEMENT AS

SELECT A.INIT_MSRMT_DATA_ID,
A.MEASR_COMP_ID,
A.D1_FROM_DTTM,
A.D1_TO_DTTM,
A.DATA_SRC_FLG,
A.TIME_ZONE_CD,
A.BUS_OBJ_CD,
A.BO_STATUS_CD,
A.BO_STATUS_REASON_CD,
A.IMD_BO_DATA_AREA,
A.STATUS_UPD_DTTM,
A.CRE_DTTM,
A.VERSION,
A.IMD_EXT_ID,
A.PREVEE_BO_DATA_AREA,
A.POSTVEE_BO_DATA_AREA,
A.TRACE_BO_DATA_AREA,
A.RAW_BO_DATA_AREA,
A.LAST_UPDATE_DTTM,
A.CRE_DTTM as ILM_DT,
'N' as ILM_ARCH_SW,
CAST(COALESCE((SELECT C.RETPERIOD
FROM D1_MEASR_COMP B, ILM_IMD_RETENTION_TMP C
WHERE B.MEASR_COMP_ID = A.MEASR_COMP_ID
AND C.MEASR_COMP_TYPE_CD = B.MEASR_COMP_TYPE_CD)
,CAST((select maint_obj_opt_val
from ci_md_mo_opt mmo
where maint_obj_cd = 'D1-IMD'
and maint_obj_opt_flg = 'FLRP'
and seq_num =
(select max(seq_num)
from ci_md_mo_opt mmo
where maint_obj_cd = 'D1-IMD'
and maint_obj_opt_flg = 'FLRP')
as NUMBER(5))
,CAST((select extractvalue( xmlparse(content fw_mcfg.mst_config_data)
,'generalMasterConfiguration/defaultRetentionPeriod')
from f1_mst_config fw_mcfg
where fw_mcfg.bus_obj_cd = 'F1-ILMMSConfig' as NUMBER(5))
, 99999) as NUMBER(5)) as RETENTION_PERIOD
5. Enable logging option for table D1_INIT_MSRMT_DATA.
   
   ```sql
   ALTER TABLE D1_INIT_MSRMT_DATA NOPARALLEL LOGGING;
   ```

6. Create Primary Index for Parent table D1_INIT_MSRMT_DATA.
   
   ```sql
   CREATE BIGFILE TABLESPACE CM_D1T304_IND DATAFILE '+DATA' SIZE 50M
   AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
   ADVANCED;
   
   CREATE UNIQUE INDEX D1T304P0 ON D1_INIT_MSRMT_DATA NOLOGGING
   PARALLEL (INIT_MSRMT_DATA_ID)
   GLOBAL PARTITION BY RANGE (INIT_MSRMT_DATA_ID) (  
   PARTITION P1 VALUES LESS THAN ('12499999999999'),  
   PARTITION P2 VALUES LESS THAN ('24999999999999'),  
   PARTITION P3 VALUES LESS THAN ('37499999999999'),  
   PARTITION P4 VALUES LESS THAN ('49999999999999'),  
   PARTITION P5 VALUES LESS THAN ('62499999999999'),  
   PARTITION P6 VALUES LESS THAN ('74999999999999'),  
   PARTITION P7 VALUES LESS THAN ('87499999999999'),  
   PARTITION P8 VALUES LESS THAN (MAXVALUE)  
   ) COMPRESS ADVANCED LOW
   
   ALTER INDEX D1T304P0 LOGGING NOPARALLEL;
   ```

7. Add Primary Key for Parent table D1_INIT_MSRMT_DATA
   
   ```sql
   ALTER TABLE D1_INIT_MSRMT_DATA ADD CONSTRAINT D1T304P0 PRIMARY
   KEY(INIT_MSRMT_DATA_ID) USING INDEX
   ```

8. Create Secondary Indexes for Parent table D1_INIT_MSRMT_DATA
   
   ```sql
   CREATE INDEX D1T304S1 ON D1_INIT_MSRMT_DATA (MEASR_COMP_ID,  
   BO_STATUS_CD, BUS_OBJ_CD, D1_TO_DTTM, D1_FROM_DTTM)
   GLOBAL PARTITION BY RANGE (MEASR_COMP_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 ( '749999999999' ),  
   PARTITION P7 VALUES LESS THAN ( '874999999999' ),  
   PARTITION P8 VALUES LESS THAN ( MAXVALUE )  
   ) COMPRESS ADVANCED LOW
   
   CREATE UNIQUE INDEX D1T304S3 ON  
   D1_INIT_MSRMT_DATA(IMD_EXT_ID,INIT_MSRMT_DATA_ID)
   GLOBAL PARTITION BY HASH(IMD_EXT_ID) (  
   PARTITION P1 TABLESPACE CM_D1T304_IND,  
   PARTITION P2 TABLESPACE CM_D1T304_IND,  
   PARTITION P3 TABLESPACE CM_D1T304_IND,  
   PARTITION P4 TABLESPACE CM_D1T304_IND,  
   PARTITION P5 TABLESPACE CM_D1T304_IND,  
   PARTITION P6 TABLESPACE CM_D1T304_IND,  
   PARTITION P7 TABLESPACE CM_D1T304_IND,  
   PARTITION P8 TABLESPACE CM_D1T304_IND  
   ) COMPRESS ADVANCED LOW
9. Create Child Tables, Primary Key, Primary Indexes and Secondary Indexes as shown below.

**Create Child Table D1_INIT_MSRMT_DATA_CHAR**

```sql
CREATE TABLE D1_INIT_MSRMT_DATA_CHAR
(
    INIT_MSRMT_DATA_ID NOT NULL ENABLE,
    CHAR_TYPE_CD NOT NULL ENABLE,
    SEQ_NUM NOT NULL ENABLE,
    CHAR_VAL DEFAULT ' ' NOT NULL ENABLE,
    ADHOC_CHAR_VAL DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK1 DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK2 DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK3 DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK4 DEFAULT ' ' NOT NULL ENABLE,
    CHAR_VAL_FK5 DEFAULT ' ' NOT NULL ENABLE,
    SRCH_CHAR_VAL DEFAULT ' ' NOT NULL ENABLE,
    VERSION DEFAULT 1 NOT NULL ENABLE,
    LAST_UPDATE_DTTM ,
    CONSTRAINT D1_INIT_MSRMT_DATA_CHAR_FK FOREIGN KEY(INIT_MSRMT_DATA_ID) REFERENCES D1_INIT_MSRMT_DATA ON DELETE CASCADE)
PARTITION BY REFERENCE (D1_INIT_MSRMT_DATA_CHAR_FK) ENABLE ROW MOVEMENT NOLOGGING PARALLEL AS SELECT /*+ PARALLEL */ * FROM ILM_D1_INIT_MSRMT_DATA_CHAR
/

ALTER TABLE D1_INIT_MSRMT_DATA_CHAR LOGGING NOPARALLEL
/

**Create Primary Index for Child Table D1_INIT_MSRMT_DATA_CHAR**

```sql
CREATE UNIQUE INDEX D1T305P0 ON D1_INIT_MSRMT_DATA_CHAR(INIT_MSRMT_DATA_ID, CHAR_TYPE_CD, SEQ_NUM) TABLESPACE CM_D1T304_IND NOLOGGING PARALLEL GLOBAL PARTITION BY RANGE(INIT_MSRMT_DATA_ID) (PARTITION P1 VALUES LESS THAN ('12499999999999'), PARTITION P2 VALUES LESS THAN ('24999999999999'), PARTITION P3 VALUES LESS THAN ('37499999999999'), PARTITION P4 VALUES LESS THAN ('49999999999999'), PARTITION P5 VALUES LESS THAN ('62499999999999'), PARTITION P6 VALUES LESS THAN ('74999999999999'), PARTITION P7 VALUES LESS THAN ('87499999999999'), PARTITION P8 VALUES LESS THAN (MAXVALUE)) COMpress ADVANCED LOW
/

ALTER INDEX D1T305P0 LOGGING NOPARALLEL
/
Create Primary Key for Child Table D1_INIT_MSRMT_DATA_CHAR

ALTER TABLE D1_INIT_MSRMT_DATA_CHAR ADD CONSTRAINT D1T305P0 PRIMARY KEY (INIT_MSRMT_DATA_ID, CHAR_TYPE_CD, SEQ_NUM) USING INDEX /

Create Secondary Indexes for Child Table D1_INIT_MSRMT_DATA_CHAR

CREATE INDEX D1T305S1 ON D1_INIT_MSRMT_DATA_CHAR(SRCH_CHAR_VAL) GLOBAL PARTITION BY HASH(SRCH_CHAR_VAL)

{ PARTITION P1 TABLESPACE CM_D1T304_IND,
  PARTITION P2 TABLESPACE CM_D1T304_IND,
  PARTITION P3 TABLESPACE CM_D1T304_IND,
  PARTITION P4 TABLESPACE CM_D1T304_IND,
  PARTITION P5 TABLESPACE CM_D1T304_IND,
  PARTITION P6 TABLESPACE CM_D1T304_IND,
  PARTITION P7 TABLESPACE CM_D1T304_IND,
  PARTITION P8 TABLESPACE CM_D1T304_IND
}
/

Create Child Table D1_INIT_MSRMT_DATA_LOG

CREATE TABLE D1_INIT_MSRMT_DATA_LOG ( INIT_MSRMT_DATA_ID NOT NULL ENABLE,
  SEQNO NOT NULL ENABLE,
  BO_STATUS_CD DEFAULT ' ' NOT NULL ENABLE,
  BO_STATUS_REASON_CD DEFAULT ' ' NOT NULL ENABLE,
  CHAR_TYPE_CD DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL DEFAULT ' ' NOT NULL ENABLE,
  ADHOC_CHAR_VAL DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK1 DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK2 DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK3 DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK4 DEFAULT ' ' NOT NULL ENABLE,
  CHAR_VAL_FK5 DEFAULT ' ' NOT NULL ENABLE,
  DESCRLONG DEFAULT ' ' NOT NULL ENABLE,
  LOG_DTTM NOT NULL ENABLE,
  LOG_ENTRY_TYPE_FLG DEFAULT ' ' NOT NULL ENABLE,
  MESSAGE_CAT_NBR DEFAULT 0 NOT NULL ENABLE,
  MESSAGE_NBR DEFAULT 0 NOT NULL ENABLE,
  USER_ID DEFAULT ' ' NOT NULL ENABLE,
  VERSION DEFAULT 1 NOT NULL ENABLE,
  LAST_UPDATE_DTTM,
  CONSTRAINT D1_INIT_MSRMT_DATA_LOG_FK FOREIGN KEY(INIT_MSRMT_DATA_ID) REFERENCES D1_INIT_MSRMT_DATA ON DELETE CASCADE)

PARTITION BY REFERENCE (D1_INIT_MSRMT_DATA_LOG_FK) ENABLE ROW MOVEMENT NOLOGGING PARALLEL
AS SELECT /*+ PARALLEL */ * FROM ILM_D1_INIT_MSRMT_DATA_LOG /

ALTER TABLE D1_INIT_MSRMT_DATA_LOG LOGGING NOPARALLEL /
Create Primary Index for Child Table D1_INIT_MSRMT_DATA_LOG
CREATE UNIQUE INDEX D1T306P0 ON
D1_INIT_MSRMT_DATA_LOG(INIT_MSRMT_DATA_ID, SEQNO)
TABLESPACE CM_D1T304_IND NOLOGGING PARALLEL
GLOBAL PARTITION BY RANGE(INIT_MSRMT_DATA_ID) (
    PARTITION P1 VALUES LESS THAN ('12499999999999'),
    PARTITION P2 VALUES LESS THAN ('24999999999999'),
    PARTITION P3 VALUES LESS THAN ('37499999999999'),
    PARTITION P4 VALUES LESS THAN ('49999999999999'),
    PARTITION P5 VALUES LESS THAN ('62499999999999'),
    PARTITION P6 VALUES LESS THAN ('74999999999999'),
    PARTITION P7 VALUES LESS THAN ('87499999999999'),
    PARTITION P8 VALUES LESS THAN (MAXVALUE)
) COMPRESS ADVANCED LOW
/
ALTER INDEX D1T306P0 LOGGING NOPARALLEL
/

Create Primary Key for Child Table D1_INIT_MSRMT_DATA_LOG
ALTER TABLE D1_INIT_MSRMT_DATA_LOG ADD CONSTRAINT D1T306P0 PRIMARY KEY (INIT_MSRMT_DATA_ID, SEQNO) USING INDEX /

Create Child Table D1_INIT_MSRMT_DATA_LOG_PARM
CREATE TABLE D1_INIT_MSRMT_DATA_LOG_PARM (
    INIT_MSRMT_DATA_ID NOT NULL ENABLE,
    SEQNO NOT NULL ENABLE,
    PARM_SEQ NOT NULL ENABLE,
    MSG_PARM_VAL DEFAULT ' ' NOT NULL ENABLE,
    MSG_PARM_TYP_FLG DEFAULT ' ' NOT NULL ENABLE,
    VERSION DEFAULT 1 NOT NULL ENABLE,
    LAST_UPDATE_DTTM ,
    CONSTRAINT D1_INIT_MSRMT_DATA_LOG_PARM_FK FOREIGN KEY(INIT_MSRMT_DATA_ID) REFERENCE D1_INIT_MSRMT_DATA ON DELETE CASCADE)
    PARTITION BY REFERENCE (D1_INIT_MSRMT_DATA_LOG_PARM_FK) ENABLE ROW MOVEMENT NOLOGGING PARALLEL
    AS SELECT /*+ PARALLEL */ * FROM ILM_D1_INIT_MSRMT_DATA_LOG_PARM /

    ALTER TABLE D1_INIT_MSRMT_DATA_LOG_PARM LOGGING NOPARALLEL /

Create Primary Index for Child Table D1_INIT_MSRMT_DATA_LOG_PARM
CREATE UNIQUE INDEX D1T307P0 ON
D1_INIT_MSRMT_DATA_LOG_PARM(INIT_MSRMT_DATA_ID, SEQNO, PARM_SEQ)
TABLESPACE CM_D1T304_IND NOLOGGING PARALLEL GLOBAL PARTITION BY RANGE(INIT_MSRMT_DATA_ID) (
    PARTITION P1 VALUES LESS THAN ( '12499999999999' ),
    PARTITION P2 VALUES LESS THAN ( '24999999999999' ),
    PARTITION P3 VALUES LESS THAN ( '37499999999999' ),
    PARTITION P4 VALUES LESS THAN ( '49999999999999' ),
    PARTITION P5 VALUES LESS THAN ( '62499999999999' ),
    PARTITION P6 VALUES LESS THAN ( '74999999999999' ),
    PARTITION P7 VALUES LESS THAN ( '87499999999999' ),
    PARTITION P8 VALUES LESS THAN ( MAXVALUE )
) COMPRESS ADVANCED LOW
/

ALTER INDEX D1T306P0 LOGGING NOPARALLEL
/

Create Primary Key for Child Table D1_INIT_MSRMT_DATA_LOG_PARM

ALTER TABLE D1_INIT_MSRMT_DATA_LOG ADD CONSTRAINT D1T307P0 PRIMARY
KEY (INIT_MSRMT_DATA_ID, SEQNO, PARM_SEQ) USING INDEX
/

Create Child Table D1_INIT_MSRMT_DATA_K

CREATE BIGFILE TABLESPACE CM_D1T314_IND DATAFILE '+DATA' SIZE 50M
AUTOEXTEND ON MAXSIZE UNLIMITED;

CREATE TABLE D1_INIT_MSRMT_DATA_K (INIT_MSRMT_DATA_ID NOT NULL ENABLE, ENV_ID NOT NULL ENABLE, CONSTRAINT D1T314P0 PRIMARY KEY (INIT_MSRMT_DATA_ID, ENV_ID) ENABLE)

ORGANIZATION INDEX
Partition by range(INIT_MSRMT_DATA_ID) {
PARTITION P1 VALUES LESS THAN ( '12499999999999' ),
PARTITION P2 VALUES LESS THAN ( '24999999999999' ),
PARTITION P3 VALUES LESS THAN ( '37499999999999' ),
PARTITION P4 VALUES LESS THAN ( '49999999999999' ),
PARTITION P5 VALUES LESS THAN ( '62499999999999' ),
PARTITION P6 VALUES LESS THAN ( '74999999999999' ),
PARTITION P7 VALUES LESS THAN ( '87499999999999' ),
PARTITION P8 VALUES LESS THAN ( MAXVALUE )
}

TABLESPACE CM_D1T314_IND
AS SELECT /*+ PARALLEL */ * FROM ILM_D1_INIT_MSRMT_DATA_K
/

ALTER TABLE D1_INIT_MSRMT_DATA_K LOGGING NOPARALLEL
/

10. After verification of the ILM based tables, the user can drop the backup “ILM” renamed tables.
Appendix H

Sample SQL for Periodic Maintenance for MDM Data

This appendix provides additional details related to creating new partitions over time as well as archiving and restoring partitions. The To Do Entry, Inbound Sync Request and Initial Measurement Data maintenance objects are used as examples.

The appendix includes the following sections:

- Adding Partition
- Archiving Partition
- Archiving Subpartition
- Restoring Partition
- Restoring Subpartition
- Compressing Partition (D1_MSRMT table only)
Adding Partition

To add a partition, follow these steps:

1. Create separate tablespace for new partition.
   
   ```sql
   CREATE BIGFILE TABLESPACE CM_XT039_P2016JAN DATAFILE '+DATA' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
   ```

2. Add partition using split operation on MAXVALUE Partition.
   
   ```sql
   ALTER TABLE CISADM.CI_TD_ENTRY SPLIT PARTITION PMAX AT (TO_DATE('2016-02-01 00:00:01','SYYYY-MM-DD HH24:MI:SS')) INTO
   (PARTITION P2016JAN TABLESPACE CM_XT039_P2016JAN, PARTITION PMAX)
   UPDATE INDEXES;
   ```

   • If the contains LOBS like F1_SYNC_REQ_IN, there will be additional statement in split partition DDL indicating tablespace on which LOB should go.

   ```sql
   ALTER TABLE CISADM.F1_SYNC_REQ_IN SPLIT PARTITION PMAX AT (TO_DATE('2016-02-01 00:00:01','SYYYY-MM-DD HH24:MI:SS')) INTO
   (PARTITION P2016JAN TABLESPACE CM_F1T191_P2016JAN, LOB(BO_DATA_AREA, POST_TRN_BO_DATA_AREA, PRE_TRN_FIN_BO_DATA_AREA, PRE_TRN_INIT_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2016JAN ), PARTITION PMAX)
   UPDATE INDEXES;
   ```

3. Enable advanced compression after SPLIT partition as it will disable the compression.

   ```sql
   ALTER TABLE CISADM.CI_TD_SRTKEY ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_MSG_PARM ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_DRLKEY ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_ENTRY_CHA ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_LOG ROW STORE COMPRESS ADVANCED;
   ```

Archiving Partition

To archive a partition, follow these steps:

1. Make the tablespace to be archived READ ONLY.

   ```sql
   ALTER TABLESPACE CM_XT039_P2011JAN READ ONLY;
   ```

2. Check the feasibility of archive using ILM_ARCH_SW = ‘N’.

   ```sql
   Select count(1) from CISADM.CI_TD_ENTRY PARTITION P2011JAN where ILM_ARCH_SW = 'N';
   ```
• IF the above query has a count of greater than ZERO records - Change the
tablespace back to read and write mode. Archive cannot be done. Do not
execute further steps. Stop archiving partition.
ALTER TABLESPACE CM_XT039_P2011JAN READ WRITE;

• IF above query has ZERO records - Archive can be performed. Continue
executing the remainder of the procedure.

3. Create separate archive tablespace for the partition that needs to be archived.
CREATE BIGFILE TABLESPACE CM_XT039_P2011JAN_ARC DATAFILE '+DATA' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;

4. Create staging tables and load data for all child tables for the MO first.
   a. CI_TD_ENTRY_CHA

      CREATE TABLE CM_XT701_P2011JAN_ARC PARALLEL NOLOGGING
      TABLESPACE CM_XT039_P2011JAN_ARC
      AS
      ( SELECT /*+ PARALLEL */ * FROM CISADM.CI_TD_ENTRY_CHA PARTITION
        (P2011JAN_S01) UNION ALL
        SELECT /*+ PARALLEL */ * FROM CI_TD_ENTRY_CHA PARTITION
        (P2011JAN_S02) UNION ALL
        SELECT /*+ PARALLEL */ * FROM CI_TD_ENTRY_CHA PARTITION
        (P2011JAN_S08) UNION ALL
        .
        .
        .
        UNION ALL
        SELECT /*+ PARALLEL */ * FROM CI_TD_ENTRY_CHA PARTITION
        (P2011JAN_S08) );
      ALTER TABLE CM_XT701_P2011JAN_ARC NOPARALLEL LOGGING;

   b. CI_TD_MSG_PARM

      CREATE TABLE CM_XT04_P2011JAN_ARC PARALLEL NOLOGGING TABLESPACE
      CM_XT039_P2011JAN_ARC
      AS
      ( SELECT /*+ PARALLEL */ * FROM CISADM.CI_TD_MSG_PARM PARTITION
        (P2011JAN_S01) UNION ALL
        SELECT /*+ PARALLEL */ * FROM CI_TD_MSG_PARM  PARTITION
        (P2011JAN_S02) UNION ALL
        SELECT /*+ PARALLEL */ * FROM CI_TD_MSG_PARM  PARTITION
        (P2011JAN_S08) UNION ALL
        .
        .
        .
        UNION ALL
        SELECT /*+ PARALLEL */ * FROM CI_TD_MSG_PARM  PARTITION
        (P2011JAN_S08) )
      )
      ALTER TABLE CM_XT04_P2011JAN_ARC NOPARALLEL LOGGING;
c. **CI_TD_LOG**

```sql
CREATE TABLE CM_XT721_P2011JAN_ARC PARALLEL NOLOGGING
   TABLESPACE CM_XT039_P2011JAN_ARC
AS
   ( SELECT /*+ PARALLEL */ * FROM CISADM.CI_TD_LOG PARTITION (P2011JAN_S01) UNION ALL
     SELECT /*+ PARALLEL */ * FROM CI_TD_LOG PARTITION (P2011JAN_S02) UNION ALL
       ...
     UNION ALL
     SELECT /*+ PARALLEL */ * FROM CI_TD_LOG PARTITION (P2011JAN_S08) );
ALTER TABLE CM_XT721_P2011JAN_ARC NOPARALLEL LOGGING;
```

d. **CI_TD_SRTKEY**

```sql
CREATE TABLE CM_XT041_P2011JAN_ARC PARALLEL NOLOGGING
   TABLESPACE CM_XT039_P2011JAN_ARC
AS
   ( SELECT /*+ PARALLEL */ * FROM CISADM.CI_TD_SRTKEY PARTITION (P2011JAN_S01) UNION ALL
     SELECT /*+ PARALLEL */ * FROM CI_TD_SRTKEY PARTITION (P2011JAN_S02) UNION ALL
       ...
     UNION ALL
     SELECT /*+ PARALLEL */ * FROM CI_TD_SRTKEY PARTITION (P2011JAN_S08) );
ALTER TABLE CM_XT041_P2011JAN_ARC NOPARALLEL LOGGING;
```

e. **CI_TD_DRLKEY**

```sql
CREATE TABLE CM_XT037_P2011JAN_ARC PARALLEL NOLOGGING
   TABLESPACE CM_XT039_P2011JAN_ARC
AS
   ( SELECT /*+ PARALLEL */ * FROM CISADM.CI_TD_DRLKEY PARTITION (P2011JAN_S01) UNION ALL
     SELECT /*+ PARALLEL */ * FROM CISADM.CI_TD_DRLKEY PARTITION (P2011JAN_S02) UNION ALL
       ...
     UNION ALL
     SELECT /*+ PARALLEL */ * FROM CISADM.CI_TD_DRLKEY PARTITION (P2011JAN_S08) );
ALTER TABLE CM_XT037_P2011JAN_ARC NOPARALLEL LOGGING;
```
5. Create staging table and load data for parent table.

```sql
CREATE TABLE CM_XT039_P2011JAN_ARC NOLOGGING PARALLEL TABLESPACE CM_XT039_P2011JAN_ARC AS
SELECT /*+ PARALLEL */ * FROM CISADM.CI_TD_ENTRY PARTITION (P2011JAN);

ALTER TABLE CM_XT039_P2011JAN_ARC NOPARALLEL LOGGING;
```


```sql
ALTER TABLESPACE CM_XT039_P2011JAN_ARC READ ONLY;

expdp system/manager DIRECTORY=DUMP_DIR DUMPFILE=CM_XT039_P2011JAN_ARC.DMP TRANSPORT_TABLESPACES = CM_XT039_P2011JAN_ARC LOGFILE=EXP_CM_XT039_P2011JAN_ARC.LOG TRANSPORT_FULL_CHECK=Y

Make sure tablespace datafile required for further import should be preserved.

<<Transport THE FILE to LOCAL DB DIRECTORY DUMP_DIR like connected to asmcmd and copied the file from cp cm_xt039_p201101_tbs_ar.553.913864937 /tugbu_perf_02/BACKUPS/test_verification/ >>
```

7. Drop the partition, partition tablespace and archive tablespace (as it is already exported).

```sql
ALTER TABLE CISADM.CI_TD_ENTRY DROP PARTITION P2011JAN UPDATE INDEXES;
DROP TABLESPACE CM_XT039_P2011JAN INCLUDING CONTENTS AND DATAFILES;
DROP TABLESPACE CM_XT039_P2011JAN_ARC INCLUDING CONTENTS AND DATAFILES;
```

---

**Archiving Subpartition**

To archive a subpartition, follow these steps:

1. Make the tablespace to be archived READ ONLY.

   ```sql
   ALTER TABLESPACE CM_D1T304_P2011JAN_S181 READ ONLY;
   ```

2. Check the feasibility of archive using `ILM_ARCH_SW = 'N'`.

   ```sql
   Select count(1) from cisadm.D1_INIT_MSRMT_DATA SUBPARTITION P2011JAN_S181 where ILM_ARCH_SW = 'N';
   ```

   IF the above query has a count of greater than ZERO records - Change the tablespace back to read and write mode. Archive cannot be done. Do not execute further steps. Stop archiving partition.

   ```sql
   ALTER TABLESPACE CM_D1T304_P2011JAN_S181 READ WRITE;
   ```

   IF the above query has ZERO records - Archive can be performed. Continue executing the remainder of the procedure.

3. Create separate archive tablespace for partition that needs to be archived.

   ```sql
   CREATE BIGFILE TABLESPACE CM_D1T304_P2011JAN_S181_ARC DATAFILE '+DATA' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
   ```
4. Create staging tables and load data for all child tables for the MO first.

```sql
CREATE TABLE CM_D1T305_P2011JAN_S181_ARC PARALLEL NOLOGGING TABLESPACE CM_D1T304_P2011JAN_S181_ARC AS
  (SELECT /*+ PARALLEL */ * FROM CISADM.D1_INIT_MSRMT_DATA_CHAR PARTITION (P2011JAN_S181));
CREATE TABLE CM_D1T306_P2011JAN_S181_ARC PARALLEL NOLOGGING TABLESPACE CM_D1T304_P2011JAN_S181_ARC AS
  (SELECT /*+ PARALLEL */ * FROM CISADM.D1_INIT_MSRMT_DATA_LOG PARTITION (P2011JAN_S181));
CREATE TABLE CM_D1T307_P2011JAN_S181_ARC PARALLEL NOLOGGING TABLESPACE CM_D1T304_P2011JAN_S181_ARC AS
  (SELECT /*+ PARALLEL */ * FROM CISADM.D1_INIT_MSRMT_DATA_LOG_PARM PARTITION (P2011JAN_S181));
ALTER TABLE CM_D1T305_P2011JAN_S181_ARC NOPARALLEL LOGGING;
ALTER TABLE CM_D1T306_P2011JAN_S181_ARC NOPARALLEL LOGGING;
ALTER TABLE CM_D1T307_P2011JAN_S181_ARC NOPARALLEL LOGGING;
```

5. Create staging table and load data for parent table

```sql
CREATE TABLE ALTERNATE TABLE CM_D1T304_P2011JAN_S181_ARC PARALLEL NOPARALLEL LOGGING; NOLOGGING PARALLEL TABLESPACE CM_D1T304_P2011JAN_S181_ARC AS
  SELECT /*+ PARALLEL */ * FROM D1_INIT_MSRMT_DATA SUBPARTITION (P2011JAN_S181);
ALTER TABLE CM_D1T304_P2011JAN_S181_ARC NOPARALLEL LOGGING;
```


```sql
ALTER TABLESPACE CM_D1T304_P2011JAN_S181_ARC READ ONLY;
```

Make sure the tablespace datafile required for future import should be preserved.

```
<<Transport THE DATAFILE to the LOCAL DB DIRECTORY DUMP_DIR. For example if connected to ascmcmd copy the file
  cp cm_d1t304_p2011jan_tbs_ar.553.913864937 /tugbu_perf_02/BACKUPS/test_verification/ >>
```

7. Drop the partition, partition tablespace and archive tablespace (since they have been exported).
ALTER TABLE D1_INIT_MSRMT_DATA DROP SUBPARTITION P2011JAN_S181
UPDATE INDEXES;
DROP TABLESPACE CM_D1T304_P2011JAN_S181 INCLUDING CONTENTS AND
DATAFILES;
DROP TABLESPACE CM_D1T304_P2011JAN_S181_ARC INCLUDING CONTENTS AND
DATAFILES;

Restoring Partition

To restore the partition, perform the following steps:

1. Create separate tablespace to restore the partition.
   CREATE BIGFILE TABLESPACE CM_XT039_P2011JAN DATAFILE '+DATA' SIZE
   50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
   ADVANCED;

2. Add partition using split operation on next greater value partition.
   ALTER TABLE CISADM.CI_TD_ENTRY SPLIT PARTITION P2011FEB AT
   (TO_DATE('2011-02-01 00:00:01','SYYYY-MM-DD HH24:MI:SS'))
   INTO
   ( PARTITION P2011JAN TABLESPACE CM_XT039_P2011JAN , PARTITION
   P2011FEB
   ) UPDATE INDEXES;

In case table contains LOBS like F1_SYNC_REQ_IN, there will be additional
statement in split partition DDL indicating tablespace on which LOB should go.

   ALTER TABLE CISADM.F1_SYNC_REQ_IN SPLIT PARTITION P2011FEB AT
   (TO_DATE('2011-02-01 00:00:01','SYYYY-MM-DD HH24:MI:SS'))
   INTO
   ( PARTITION P2011JAN TABLESPACE CM_F1T191_P2011JAN
   LOB(BO_DATA_AREA,PRE_TRN_INIT_BO_DATA_AREA,PRE_TRN_FIN_BO_DATA_AREA,
   POST_TRN_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW
   COMPRESS MEDIUM CACHE TABLESPACE CM_F1T191_P2011JAN ), PARTITION
   P2011FEB
   ) UPDATE INDEXES;

3. Enable advanced compression after SPLIT partition as it will disable the
   compression.
   ALTER TABLE CISADM.CI_TD_SRTKEY ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_MSG_PARM ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_DRLKEY ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_ENTRY_CHA ROW STORE COMPRESS ADVANCED;
   ALTER TABLE CISADM.CI_TD_LOG ROW STORE COMPRESS ADVANCED;

4. Import tablespace using TRANSPORT_TABLESPACES method.
   impdp system/manager DIRECTORY=DUMP_DIR
   DUMPFILE=CM_D1T304_P2011JAN_S181_ARC.DMP
   PARTITION_OPTIONS=DEPARTITION
   LOGFILE=IMP_CM_D1T304_P2011JAN_S181_ARC.LOG TRANSPORT_DATAFILES=/
tugbu_perf_02/BACKUPS/test_verification/
   cm_d1t304_p2011jan_tbs_ar.553.913864937
5. Load data into parent table first from the staging table.

```
ALTER SESSION ENABLE PARALLEL DML;
INSERT /*+ APPEND PARALLEL */ INTO CISADM.CI_TD_ENTRY SELECT /*+ PARALLEL */ * FROM CM_XT039_P2011JAN_ARC;
COMMIT;
```

6. Load data into child table from the staging table.

```
For each Child IN LIST OF CHILD TABLES, perform the following:
INSERT /*+ APPEND PARALLEL */ INTO CISADM.CI_TD_ENTRY_CHA SELECT /*+ PARALLEL */ * FROM CM_XT701_P2011JAN_ARC;
COMMIT;
INSERT /*+ APPEND PARALLEL */ INTO CISADM.CI_TD_MSG_PARM SELECT /*+ PARALLEL */ * FROM CM_XT04_P2011JAN_ARC;
COMMIT;
INSERT /*+ APPEND PARALLEL */ INTO CISADM.CI_TD_LOG SELECT /*+ PARALLEL */ * FROM CM_XT721_P2011JAN_ARC;
COMMIT;
INSERT /*+ APPEND PARALLEL */ INTO CISADM.CI_TD_SRTKEY SELECT /*+ PARALLEL */ * FROM CM_XT041_P2011JAN_ARC;
COMMIT;
INSERT /*+ APPEND PARALLEL */ INTO CISADM.CI_TD_DRLKEY SELECT /*+ PARALLEL */ * FROM CM_XT037_P2011JAN_ARC;
COMMIT;
```

7. Drop the archive tablespace after import is import and data loading is successful.

```
DROP TABLESPACE CM_XT039_P2011JAN_ARC INCLUDING CONTENTS AND DATAFILES;
```

---

**Restoring Subpartition**

To restore the subpartition, follow these steps:

1. Create separate tablespace to restore the partition.

```
CREATE BIGFILE TABLESPACE CM_D1T304_P2011JAN_S181 DATAFILE 'DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
```

2. Add partition using split operation on next greater value partition.

```
ALTER TABLE CISADM.D1_INIT_MSRMT_DATA SPLIT SUBPARTITION P2011JAN_SMAX AT (181)
INTO
(SUBPARTITION P2011JAN_S181 TABLESPACE CM_D1T304_P2011JAN_S181
LOB(IMD_BO_DATA_AREA, PREVEE_BO_DATA_AREA, POSTVEE_BO_DATA_AREA, TRACE_BO_DATA_AREA, RAW_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE TABLESPACE CM_D1T304_P2011JAN_S181)
, SUBPARTITION P2011JAN_SMAX) UPDATE INDEXES;
```
3. Enable advanced compression after SPLIT partition as it will disable the compression.

   ALTER TABLE D1_INIT_MSRMT_DATA_CHAR ROW STORE COMPRESS ADVANCED;
   ALTER TABLE D1_INIT_MSRMT_DATA_LOG ROW STORE COMPRESS ADVANCED;
   ALTER TABLE D1_INIT_MSRMT_DATA_LOG_PARM ROW STORE COMPRESS ADVANCED;

4. Import tablespace using TRANSPORT_TABLESPACES method.

   impdp system/manager DIRECTORY=DUMP_DIR
   DUMPFILE=CM_D1T304_P2011JAN_S181_ARC.DMP
   PARTITION_OPTIONS=DEPARTITION
   LOGFILE=IMP_CM_D1T304_P2011JAN_S181_ARC.LOG
   TRANSPORT_DATAFILES=/
tugbu_perf_02/BACKUPS/test_verification/
   cm_d1t304_p2011tbs_ar.553.913864937

5. Load data into parent table first from the staging table.

   ALTER SESSION ENABLE PARALLEL DML;

   INSERT /*+ APPEND PARALLEL */ INTO CISADM.D1_INIT_MSRMT_DATA
   SELECT /*+ PARALLEL */ * FROM CM_D1T304_P2011JAN_S181_ARC;
   COMMIT;

6. Load data into child table from the staging table.

   For each Child IN LIST OF CHILD TABLES, perform the following:

   INSERT /*+ APPEND PARALLEL */ INTO D1_INIT_MSRMT_DATA_CHAR
   SELECT /*+ PARALLEL */ * FROM CM_D1T305_P2011JAN_S181_ARC;
   COMMIT;

   INSERT /*+ APPEND PARALLEL */ INTO D1_INIT_MSRMT_DATA_LOG
   SELECT /*+ PARALLEL */ * FROM CM_D1T306_P2011JAN_S181_ARC;
   COMMIT;

   INSERT /*+ APPEND PARALLEL */ INTO D1_INIT_MSRMT_DATA_LOG_PARM
   SELECT /*+ PARALLEL */ * FROM CM_D1T307_P2011JAN_S181_ARC;
   COMMIT;

7. Drop the archive tablespace after import is import and data loading is successful.

   DROP TABLESPACE CM_D1T304_P2011JAN_S181_ARC INCLUDING CONTENTS AND
   DATAFILES;

---

**Compressing Partition (D1_MSRMT table only)**

To compress a partition, perform the steps below:

1. Create Compressed Partition Tablespace.

   CREATE BIGFILE TABLESPACE CM_D1T298_P2011JAN_C DATAFILE '+DATADG'
   SIZE 50M
   AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS
   ADVANCED;

   **Note:** Perform Steps 2 - 9 for each subpartition (S01 – SMAX)
2. Create and Load Data Into Staging Table.
   
   ```sql
   CREATE TABLE D1_MSRMT_P2011JAN_S01 PARALLEL NOLOGGING TABLESPACE CM_D1T298_P2011JAN_C
   AS
   SELECT /*+ PARALLEL */ * FROM D1_MSRMT SUBPARTITION(P2011JAN_S01)
   ORDER BY MEASR_COMP_ID, MSRMT_DTTM;
   ```

3. Enable Logging on Newly Created Staging Table.
   
   ```sql
   ALTER TABLE D1_MSRMT_P2011JAN_S01 NOPARALLEL LOGGING;
   ```

4. Create Primary Unique Index on Staging Table.
   
   ```sql
   CREATE UNIQUE INDEX D1T298P0_P2011JAN_S01
   ON D1_MSRMT_P2011JAN_S01(MEASR_COMP_ID, MSRMT_DTTM)
   PARALLEL NOLOGGING COMPRESS ADVANCED LOW TABLESPACE CM_D1T298_P2011JAN_C;
   ```

5. Create Primary Key Constraint on Staging Table.
   
   ```sql
   ALTER TABLE D1_MSRMT_P2011JAN_S01 ADD CONSTRAINT
   D1T298P0_P2011JAN_S01 PRIMARY KEY(MEASR_COMP_ID, MSRMT_DTTM) USING
   INDEX;
   ```

6. Enable Logging on Primary Key Index.
   
   ```sql
   ALTER INDEX D1T298P0_P2011JAN_S01 NOPARALLEL LOGGING;
   ```

7. Exchange D1_MSRMT Table Subpartition With Newly Created Staging Table.
   
   ```sql
   ALTER TABLE D1_MSRMT EXCHANGE SUBPARTITION(P2011JAN_S01) WITH TABLE
   D1_MSRMT_P2011JAN_S01 INCLUDING INDEXES;
   ```

   **Note:** Ensure that steps 2-9 have been executed for each subpartition (S01 – SMAX) before continuing.

8. Drop Original Uncompressed Tablespace.
   
   ```sql
   DROP TABLESPACE CM_D1T298_P2011JAN INCLUDING CONTENTS AND
   DATAFILES;
   ```

   
   ```sql
   ALTER TABLE D1_MSRMT MODIFY DEFAULT ATTRIBUTES FOR PARTITION
   P2011JAN TABLESPACE CM_D1T298_P2011JAN_C;
   ```

10. Rename Tablespace to Original Tablespace Name.
    
    ```sql
    ALTER TABLESPACE CM_D1T298_P2011JAN_C RENAME TO CM_D1T298_P2011JAN;
    ```
In this release of Oracle Utilities Customer To Meter, there are enhancements that change the behavior of some objects. Upgrading customers may wish to update existing data to align with how newly created data will behave.

The following sample scripts are provided to help upgrading clients. Customers are advised to review, edit, and test these sample script to ensure they meet business and data requirement.

All scripts are delivered with logic to run in threads. In the format that the script is delivered, these scripts will not process any records. The scripts must be edited to set specific key ranges so that it can be run in parallel or process all records.

This appendix consists of the following:

- Updating Customer Contact Account and Premise
- Updating Preferred Contact Method on Legacy Values
Updating Customer Contact Account and Premise

In this release of Oracle Utilities Customer To Meter, the customer contact capability has been enhanced so that customer contacts can be linked to a person, account and/or premise. This section contains sample SQL that upgrading implementations can use to update the account ID and premise ID fields on existing customer contacts.

All Oracle Utilities Customer To Meter system processes that create customer contacts have been enhanced to populate account and/or premise. Not all of these areas retain a link to the customer contact created and the SQL provided is just a sample.

These scripts update Customer Contact Account and Premise from the following sources:

- Customer Contact Characteristics
- Collection Events
- Severance Events
- Write off Events
- Overdue Process Logs

The other Oracle Utilities Customer To Meter process enhanced are algorithm types delivered for the below plug-in spots. Examine your organization’s customer contacts to determine if you should update customer contacts created from the following:

- FA Remark Algorithm
- Meter read Remark Algorithm
- Case Type Enter Status Algorithm
- Customer Class Order Completion
- SA Type SA Stop
- SA Type - SA Activation
- Service Credit Membership Type - Membership Activation
- Service Credit Membership Type - Membership Creation
- Campaign - Order Completion
- Lead Event Type (BO) - Lead Event Completion
- Notification Type (BO) - Create Notification

Since existing customer contact records are all person-based, all account updates should occur before premise updates to ensure the premise is associated with the account.

These are intended to be run in the following order:

1. Update Account ID from characteristic.

Note the following about this SQL:

- This cannot handle two characteristic types that are both linked to Acct FK Ref with different char values. If this exists such as FROM_ACCT and TO_ACCT with different values, this will not update.
- Limited to accounts linked to CC person to not violate new CC validations.
• Change FK Ref if your implementation has introduced new FK Ref values for ACCT.

```
UPDATE CI_CC X
SET X.ACCT_ID =
  (SELECT DISTINCT(a.CHAR_VAL_FK1)
   FROM CI_CC_CHAR A,
       CI_CHAR_TYPE B,
       CI_CC C
   WHERE a.CHAR_TYPE_CD = B.CHAR_TYPE_CD
   AND A.CC_ID = C.CC_ID
   AND B.FK_REF_CD <> ' '
   AND b.fk_ref_cd = 'ACCT'
   AND a.CC_ID = X.CC_ID
   AND EXISTS
     (SELECT 'x'
      FROM CI_ACCT_PER D
      WHERE D.PER_ID = C.PER_ID
      AND D.ACCT_ID = a.CHAR_VAL_FK1)
  )
WHERE CC_ID =
  (SELECT A.CC_ID
   FROM CI_CC_CHAR A,
       CI_CHAR_TYPE B,
       CI_CC C
   WHERE a.CHAR_TYPE_CD = B.CHAR_TYPE_CD
   AND A.CC_ID = C.CC_ID
   AND B.FK_REF_CD <> ' '
   AND b.fk_ref_cd = 'ACCT'
   AND a.CC_ID = X.CC_ID
   AND EXISTS
     (SELECT 'x'
      FROM CI_ACCT_PER D
      WHERE D.PER_ID = C.PER_ID
      AND D.ACCT_ID = a.CHAR_VAL_FK1)
  )
AND 1 =
  (SELECT COUNT (DISTINCT a.CHAR_VAL_FK1)
   FROM CI_CC_CHAR A,
       CI_CHAR_TYPE B,
       CI_CC C
   WHERE a.CHAR_TYPE_CD = B.CHAR_TYPE_CD
   AND A.CC_ID = C.CC_ID
   AND B.FK_REF_CD <> ' '
   AND b.fk_ref_cd = 'ACCT'
   AND a.CC_ID = X.CC_ID
  )
AND X.ACCT_ID IS NULL
--AND X.CC_ID BETWEEN '0000000000' AND '9999999999';
AND X.CC_ID BETWEEN '0000000000' AND '0000000000';
```

2. Update Account ID from Coll event.

```
UPDATE CI_CC X
SET X.ACCT_ID =
  (SELECT a.acct_id
   FROM CI_COLL_PROC A,
       CI_COLL_EVT_CC C
   WHERE a.coll_proc_id = c.coll_proc_ID
   AND c.cc_id = x.cc_id
```

WHERE X.CC_ID IN
  (SELECT CC_ID FROM CI_COLL_EVT_CC
   )
AND X.ACCT_ID IS NULL
  --AND X.CC_ID BETWEEN '0000000000' AND '9999999999';
  AND X.CC_ID BETWEEN '0000000000' AND '0000000000';

3. Update Account ID from Sev event.

   UPDATE CI_CC X
   SET X.ACCT_ID =
     (SELECT a.acct_id
      FROM CI_SA A,
      CI_SEV_PROC B,
      CI_SEV_EVT_CC C
      WHERE a.SA_ID = B.SA_ID
      AND B.sev_proc_id = c.sev_proc_id
      AND c.cc_id = x.cc_id
     )
   WHERE X.CC_ID IN
     (SELECT CC_ID FROM CI_SEV_EVT_CC
     )
   AND X.ACCT_ID IS NULL
     --AND X.CC_ID BETWEEN '0000000000' AND '9999999999';
     AND X.CC_ID BETWEEN '0000000000' AND '0000000000';

4. Update Premise ID from Sev event.

   Using premise linked to SP linked to FA linked to same Sev Proc First.

   UPDATE CI_CC X
   SET X.PREM_ID =
     (SELECT DISTINCT h.prem_id
      FROM CI_SEV_EVT_CC D,
      CI_SEV_EVT_FA E,
      CI_FA G,
      CI_SP H
      WHERE D.SEV_PROC_ID = E.SEV_PROC_ID
      AND e.fa_id = g.fa_id
      AND g.sp_id = h.sp_id
      AND d.cc_id = x.cc_id
     )
   WHERE X.CC_ID IN
     (SELECT CC_ID
      FROM CI_SEV_EVT_CC D,
      CI_SEV_EVT_FA E
      WHERE D.SEV_PROC_ID = E.SEV_PROC_ID
     )
   AND X.PREM_ID IS NULL
     --AND X.CC_ID BETWEEN '0000000000' AND '9999999999';
     AND X.CC_ID BETWEEN '0000000000' AND '0000000000';

5. Update Premise ID from Sev event.

   Using premise linked to SA via Char Premise second.

   Note: There is no SQL that will try to find a premise via SASP. The likelihood of more than one premise is too high.

   UPDATE CI_CC X
   SET X.PREM_ID =
     (SELECT NVL(a.char_prem_id,NULL)
      WHERE X.CC_ID IN
        (SELECT CC_ID FROM CI_COLL_EVT_CC
        )
      AND X.ACCT_ID IS NULL
        --AND X.CC_ID BETWEEN '0000000000' AND '9999999999';
        AND X.CC_ID BETWEEN '0000000000' AND '0000000000';
   )
FROM CI_SA A, 
   CI_SEV_PROC B, 
   CI_SEV_EVT_CC C 
WHERE a.SA_ID = B.SA_ID 
AND B.sev_proc_id = c.sev_proc_id 
AND A.CHAR_PREM_ID <> '' 
AND c.cc_id = x.cc_id 
WHERE X.CC_ID IN 
   (SELECT CC_ID FROM CI_SEV_EVT_CC 
   ) 
AND X.PREM_ID IS NULL 
   --AND X.CC_ID BETWEEN '0000000000' AND '9999999999'; 
AND X.CC_ID BETWEEN '0000000000' AND '0000000000';

6. Update Account ID from WO event.

   UPDATE CI_CC X 
   SET X.ACCT_ID = 
      (SELECT a.acct_id 
       FROM CI_WO_PROC A, 
       CI_WO_EVT_CC C 
       WHERE a.wo_proc_id = c.wo_proc_ID 
       AND c.cc_id = x.cc_id 
       ) 
   WHERE X.CC_ID IN 
      (SELECT CC_ID FROM CI_WO_EVT_CC 
      ) 
   AND X.ACCT_ID IS NULL 
      --AND X.CC_ID BETWEEN '0000000000' AND '9999999999'; 
   AND X.CC_ID BETWEEN '0000000000' AND '0000000000';


   NOTE: You must hardcode the Char Type Code your implementation has introduced.

   UPDATE CI_CC X 
   SET X.ACCT_ID = 
      (SELECT b.acct_id 
       FROM CI_OD_PROC_LOG A, 
       CI_OD_PROC B 
       WHERE a.char_type_cd = 'CCID' 
       AND a.OD_PROC_ID = B.OD_PROC_ID 
       AND trim(a.char_val_fk1) = x.cc_id 
       ) 
   WHERE X.CC_ID IN 
      (SELECT a.char_val_fk1 
       FROM CI_OD_PROC_LOG A, 
       CI_OD_PROC B 
       WHERE a.char_type_cd = 'CCID' 
       AND a.OD_PROC_ID = B.OD_PROC_ID 
       ) 
   AND X.ACCT_ID IS NULL 
      --AND X.CC_ID BETWEEN '0000000000' AND '9999999999'; 
   AND X.CC_ID BETWEEN '0000000000' AND '0000000000';

8. Update Premise ID from CC Char.

   Some notes about this SQL:
• This cannot handle two characteristic types that are both linked to Prem FK Ref with different char values. If this exists such as OLD_PREM and NEW_PREM with different values, this will not update.

• Limited to premise associated with account if acct is populated on the CC to not violate new CC validations.

• Change FK Ref if your implementation has introduced new FK Ref values for PREM.

    UPDATE CI_CC X
    SET X.PREM_ID =
        (SELECT DISTINCT(a.CHAR_VAL_FK1)
         FROM CI_CC_CHAR A,
            CI_CHAR_TYPE B,
            CI_CC C
         WHERE a.CHAR_TYPE_CD = B.CHAR_TYPE_CD
         AND A.CC_ID = C.CC_ID
         AND B.FK_REF_CD <> ' '  
         AND b.fk_ref_cd = 'PREM'
         AND a.CC_ID = X.CC_ID
         AND (C.ACCT_ID IS NULL
         OR C.ACCT_ID IS NOT NULL
         AND a.CHAR_VAL_FK1 IN
            (SELECT E.char_prem_id FROM CI_SA E WHERE E.ACCT_ID = C.ACCT_ID
             UNION
            SELECT H.PREM_ID
            FROM CI_SA F,
                CI_SA_SP G,
                CI_SP H
            WHERE f.ACCT_ID = C.ACCT_ID
            AND F.SA_ID = G.SA_ID
            AND G.SP_ID = H.SP_ID
            )
        )
    )
    WHERE CC_ID =
        (SELECT A.CC_ID
         FROM CI_CC_CHAR A,
            CI_CHAR_TYPE B,
            CI_CC C
         WHERE a.CHAR_TYPE_CD = B.CHAR_TYPE_CD
         AND A.CC_ID = C.CC_ID
         AND B.FK_REF_CD <> ' '  
         AND b.fk_ref_cd = 'PREM'
         AND a.CC_ID = X.CC_ID
         AND (C.ACCT_ID IS NULL
         OR C.ACCT_ID IS NOT NULL
         AND a.CHAR_VAL_FK1 IN
            (SELECT E.char_prem_id FROM CI_SA E WHERE E.ACCT_ID = C.ACCT_ID
             UNION
            SELECT H.PREM_ID
            FROM CI_SA F,
                CI_SA_SP G,
                CI_SP H
            WHERE f.ACCT_ID = C.ACCT_ID
            AND F.SA_ID = G.SA_ID
            AND G.SP_ID = H.SP_ID
            )
        )
    )

Updating Preferred Contact Method on Legacy Values

In this release of Oracle Utilities Customer To Meter, a feature is introduced that some system processes use to determine if person phone and email or person contacts is being used.

If moving to using person contacts, some contact method values on case and customer contact are suppressed and existing records show as <invalid value>.

These scripts update the preferred contact method on cases and customer contacts from legacy values to person contact.

1. Update Case Preferred Contact Method from Email to Primary Email Person Contact:

   ```sql
   UPDATE CI_CASE X
   SET X.CONTACT_METH_FLG = 'C1PC',
       X.C1_CONTACT_ID = (SELECT B.C1_CONTACT_ID
                        FROM CI_CASE A, CISADM.C1_PER_CONTDET B, CISADM.C1_COMM_RTE_TYPE C
                        WHERE CONTACT_METH_FLG  = 'EM'
                        AND A.CONTACT_PER_ID    = B.PER_ID
                        AND B.COMM_RTE_TYPE_CD  = C.COMM_RTE_TYPE_CD
                        AND B.CND_PRIMARY_FLG   = 'C1YS'
                        AND C.COMM_RTE_METH_FLG = 'EMAIL'
                        AND X.CASE_ID = A.CASE_ID)
   WHERE X.CASE_ID IN (SELECT A.CASE_ID
                        FROM CI_CASE A, CISADM.C1_PER_CONTDET B, CISADM.C1_COMM_RTE_TYPE C
                        WHERE CONTACT_METH_FLG  = 'EM'
                        AND A.CONTACT_PER_ID    = B.PER_ID
                        AND B.COMM_RTE_TYPE_CD  = C.COMM_RTE_TYPE_CD
                        AND B.CND_PRIMARY_FLG   = 'C1YS'
                        AND C.COMM_RTE_METH_FLG = 'EMAIL'
                        )
   --AND X.CASE_ID BETWEEN '0000000000' AND '9999999999';
   AND X.CASE_ID BETWEEN '0000000000' AND '0000000000';
   ```

2. Update Case Preferred Contact Method from Fax to Primary Fax Person Contact:

   ```sql
   UPDATE CI_CASE X
   SET X.CONTACT_METH_FLG = 'C1PC',
       X.C1_CONTACT_ID = (SELECT B.C1_CONTACT_ID
                        FROM CI_CASE A, CISADM.C1_PER_CONTDET B, CISADM.C1_COMM_RTE_TYPE C
                        WHERE CONTACT_METH_FLG  = 'EM'
                        AND A.CONTACT_PER_ID    = B.PER_ID
                        AND B.COMM_RTE_TYPE_CD  = C.COMM_RTE_TYPE_CD
                        AND B.CND_PRIMARY_FLG   = 'C1YS'
                        AND C.COMM_RTE_METH_FLG = 'EMAIL'
                        )
   WHERE X.CASE_ID IN (SELECT A.CASE_ID
                        FROM CI_CASE A, CISADM.C1_PER_CONTDET B, CISADM.C1_COMM_RTE_TYPE C
                        WHERE CONTACT_METH_FLG  = 'EM'
                        AND A.CONTACT_PER_ID    = B.PER_ID
                        AND B.COMM_RTE_TYPE_CD  = C.COMM_RTE_TYPE_CD
                        AND B.CND_PRIMARY_FLG   = 'C1YS'
                        AND C.COMM_RTE_METH_FLG = 'EMAIL'
                        )
   --AND X.CASE_ID BETWEEN '0000000000' AND '9999999999';
   AND X.CASE_ID BETWEEN '0000000000' AND '0000000000';
   ```
Updating Preferred Contact Method on Legacy Values

SAMPLE SCRIPTS FOR CUSTOMER CONTACT ENHANCEMENT

I - 8

ORACLE UTILITIES CUSTOMER TO METER
DATABASE ADMINISTRATOR'S GUIDE

3. Update Case Preferred Contact Method from Phone to Primary Phone Person Contact:

UPDATE CI_CASE X
SET X.CONTACT_METH_FLG = 'C1PC',
X.C1_CONTACT_ID =
(SELECT B.C1_CONTACT_ID
FROM CI_CASE A,
    CISADM.C1_PER_CONTDET B,
    CISADM.C1_COMM_RTE_TYPE C
WHERE CONTACT_METH_FLG = 'PH'
AND A.CONTACT_PER_ID = B.PER_ID
AND B.COMM_RTE_TYPE_CD = C.COMM_RTE_TYPE_CD
AND B.CND_PRIMARY_FLG = 'C1YS'
AND C.COMM_RTE_METH_FLG = 'PHONE'
AND X.CASE_ID = A.CASE_ID
)
WHERE X.CASE_ID IN
(SELECT A.CASE_ID
FROM CI_CASE A,
    CISADM.C1_PER_CONTDET B,
    CISADM.C1_COMM_RTE_TYPE C
WHERE CONTACT_METH_FLG = 'PH'
AND A.CONTACT_PER_ID = B.PER_ID
AND B.COMM_RTE_TYPE_CD = C.COMM_RTE_TYPE_CD
AND B.CND_PRIMARY_FLG = 'C1YS'
AND C.COMM_RTE_METH_FLG = 'PHONE'
)
--AND X.CASE_ID BETWEEN '0000000000' AND '9999999999';
AND X.CASE_ID BETWEEN '0000000000' AND '0000000000';

4. Update Customer Contact Preferred Contact Method from Email to Primary Email Person Contact:

UPDATE CI_CC X
SET X.CONTACT_METH_FLG = 'C1PC',
X.C1_CONTACT_ID =
(SELECT B.C1_CONTACT_ID
FROM CI_CASE A,
    CISADM.C1_PER_CONTDET B,
    CISADM.C1_COMM_RTE_TYPE C
WHERE CONTACT_METH_FLG = 'EMAIL'
AND A.CONTACT_PER_ID = B.PER_ID
AND B.COMM_RTE_TYPE_CD = C.COMM_RTE_TYPE_CD
AND B.CND_PRIMARY_FLG = 'C1YS'
AND C.COMM_RTE_METH_FLG = 'EMAIL'
AND X.CASE_ID = A.CASE_ID
)
WHERE X.CASE_ID IN
(SELECT A.CASE_ID
FROM CI_CASE A,
    CISADM.C1_PER_CONTDET B,
    CISADM.C1_COMM_RTE_TYPE C
WHERE CONTACT_METH_FLG = 'EMAIL'
AND A.CONTACT_PER_ID = B.PER_ID
AND B.COMM_RTE_TYPE_CD = C.COMM_RTE_TYPE_CD
AND B.CND_PRIMARY_FLG = 'C1YS'
AND C.COMM_RTE_METH_FLG = 'EMAIL'
)
--AND X.CASE_ID BETWEEN '0000000000' AND '9999999999';
AND X.CASE_ID BETWEEN '0000000000' AND '0000000000';
(SELECT B.C1_CONTACT_ID
FROM CI_CC A,
    CISADM.C1_PER_CONTDET B,
    CISADM.C1_COMM_RTE_TYPE C
WHERE CONTACT_METH_FLG = 'EM'
AND A.PER_ID = B.PER_ID
AND B.COMM_RTE_TYPE_CD = C.COMM_RTE_TYPE_CD
AND B.CND_PRIMARY_FLG = 'C1YS'
AND C.COMM_RTE_METH_FLG = 'EMAIL'
AND X.CC_ID = A.CC_ID
)
WHERE X.CC_ID IN
  (SELECT A.CC_ID
FROM CI_CC A,
    CISADM.C1_PER_CONTDET B,
    CISADM.C1_COMM_RTE_TYPE C
WHERE CONTACT_METH_FLG = 'EM'
AND A.PER_ID = B.PER_ID
AND B.COMM_RTE_TYPE_CD = C.COMM_RTE_TYPE_CD
AND B.CND_PRIMARY_FLG = 'C1YS'
AND C.COMM_RTE_METH_FLG = 'EMAIL'
)
--AND X.CC_ID BETWEEN '0000000000' AND '9999999999';
AND X.CC_ID BETWEEN '0000000000' AND '0000000000';

5. Update Customer Contact Preferred Contact Method from Fax to Primary Fax Person Contact:

UPDATE CI_CC X
SET X.CONTACT_METH_FLG = 'C1PC',
    X.C1_CONTACT_ID =
  (SELECT B.C1_CONTACT_ID
FROM CI_CC A,
    CISADM.C1_PER_CONTDET B,
    CISADM.C1_COMM_RTE_TYPE C
WHERE CONTACT_METH_FLG = 'FAX'
AND A.PER_ID = B.PER_ID
AND B.COMM_RTE_TYPE_CD = C.COMM_RTE_TYPE_CD
AND B.CND_PRIMARY_FLG = 'C1YS'
AND C.COMM_RTE_METH_FLG = 'FAX'
AND X.CC_ID = A.CC_ID
)
WHERE X.CC_ID IN
  (SELECT A.CC_ID
FROM CI_CC A,
    CISADM.C1_PER_CONTDET B,
    CISADM.C1_COMM_RTE_TYPE C
WHERE CONTACT_METH_FLG = 'FAX'
AND A.PER_ID = B.PER_ID
AND B.COMM_RTE_TYPE_CD = C.COMM_RTE_TYPE_CD
AND B.CND_PRIMARY_FLG = 'C1YS'
AND C.COMM_RTE_METH_FLG = 'FAX'
)
--AND X.CC_ID BETWEEN '0000000000' AND '9999999999';
AND X.CC_ID BETWEEN '0000000000' AND '0000000000';

6. Update Customer Contact Preferred Contact Method from Phone to Primary Phone Person Contact:

UPDATE CI_CC X
SET X.CONTACT_METH_FLG = 'C1PC',
    X.C1_CONTACT_ID =
Updating Preferred Contact Method on Legacy Values

Sample Scripts for Customer Contact Enhancement

Oracle Utilities Customer To Meter Database Administrator’s Guide

7. The following shows remaining cases that need to be investigated manually:

```sql
SELECT *
FROM CI_CASE
WHERE CONTACT_METH_FLG <> ' ' 
AND CONTACT_METH_FLG NOT IN ('N/A','POST','C1PC');
```

8. The following shows remaining Customer Contacts that need to be investigated manually:

```sql
SELECT *
FROM CI_CC
WHERE CONTACT_METH_FLG <> ' ' 
AND CONTACT_METH_FLG NOT IN ('N/A','POST','C1PC');
```
This section specifies the partitioning and compression strategies recommended for an initial Oracle Utilities Meter Data Management database configuration. It includes the following topics:

- Partitioning Recommendations
- Compression Recommendations

Note: If Information Lifecycle Management is part of your implementation, please refer to the chapter Information Lifecycle Management and Data Archiving in MDM in this guide for instructions on partitioning objects when using ILM.
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:

- The Table Partitioning scheme for Transaction tables is focused primarily on tables associated with Measurement MO, Measurement Log MO and Initial-Measurement-Data MO.

- `D1_MSRMT`, `D1_MSRMT_CHAR`, `D1_MSRMT_LOG`, `D1_MSRMT_LOG_PARM` tables can be partitioned by `MSRMT_DTTM`. Bi-monthly partitions is a good start. Subpartition these tables by `MEASR_COMP_ID` (8 subpartitions should be a good number to start with).

- `D1_INIT_MSRMT_DATA` table can be partitioned by `D1_TO_DTTM`. Bi-monthly partitions is a good start. Subpartition `D1_INIT_MSRMT_DATA` table by `MEASR_COMP_ID` (8 subpartitions should be a good number to start with).

- `D1_INIT_MSRMT_DATA_CHAR`, `D1_INIT_MSRMT_DATA_LOG`, `D1_INIT_MSRMT_DATA_LOG_PARM` tables are reference partitioned to the parent table.

- `D1_INIT_MSRMT_DATA_K` table can be partitioned by `INIT_MSRMT_DATA_ID` (8 sub partitions should be a good number to start with).

The following sections gives partition recommendation and can be used as reference. Create one tablespace per partition as needed. It includes the following:

- `D1_MSRMT`
- `D1_MSRMT_CHAR`
- `D1_MSRMT_LOG`
- `D1_MSRMT_LOG_PARM`
- `D1_INIT_MSRMT_DATA`
- `D1_INIT_MSRMT_DATA_CHAR`
- `D1_INIT_MSRMT_DATA_K`
- `D1_INIT_MSRMT_DATA_LOG`
- `D1_INIT_MSRMT_DATA_LOG_PARM`

### D1_MSRMT

```sql
CREATE BIGFILE TABLESPACE CM_D1T298_P2011JAN DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T298_P2011MAR DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T298_P2011MAY DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T298_P2011JUL DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T298_P2011SEP DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T298_P2011NOV DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T298_PMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
```
CREATE TABLE D1_MSRMT(
    MEASR_COMP_ID CHAR(12) NOT NULL ENABLE,
    MSRMT_DTTM DATE NOT NULL ENABLE,
    BO_STATUS_CD CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
    MSRMT_COND_FLG CHAR(6 BYTE) DEFAULT ' ' NOT NULL ENABLE,
    MSRMT_USE_FLG CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
    MSRMT_LOCAL_DTTM DATE NOT NULL ENABLE,
    MSRMT_VAL NUMBER(16,6) DEFAULT 0 NOT NULL ENABLE,
    ORIG_INIT_MSRMT_ID CHAR(14) DEFAULT ' ' NOT NULL ENABLE,
    PREV_MSRMT_DTTM DATE,
    MSRMT_VAL1 NUMBER(16,6) DEFAULT 0 NOT NULL ENABLE,
    MSRMT_VAL2 NUMBER(16,6) DEFAULT 0 NOT NULL ENABLE,
    MSRMT_VAL3 NUMBER(16,6) DEFAULT 0 NOT NULL ENABLE,
    MSRMT_VAL4 NUMBER(16,6) DEFAULT 0 NOT NULL ENABLE,
    MSRMT_VAL5 NUMBER(16,6) DEFAULT 0 NOT NULL ENABLE,
    MSRMT_VAL6 NUMBER(16,6) DEFAULT 0 NOT NULL ENABLE,
    MSRMT_VAL7 NUMBER(16,6) DEFAULT 0 NOT NULL ENABLE,
    MSRMT_VAL8 NUMBER(16,6) DEFAULT 0 NOT NULL ENABLE,
    MSRMT_VAL9 NUMBER(16,6) DEFAULT 0 NOT NULL ENABLE,
    MSRMT_VAL10 NUMBER(16,6) DEFAULT 0 NOT NULL ENABLE,
    BUS_OBJ_CD CHAR(30) DEFAULT ' ' NOT NULL ENABLE,
    CRE_DTTM DATE NOT NULL ENABLE,
    STATUS_UPD_DTTM DATE NOT NULL ENABLE,
    USER_EDITED_FLG CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
    VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    LAST_UPDATE_DTTM DATE,
    READING_VAL NUMBER(16,6),
    COMBINED_MULTIPLIER NUMBER(12,6),
    READING_COND_FLG CHAR(6)
) ENABLE ROW MOVEMENT
PARTITION BY RANGE (MSRMT_DTTM) SUBPARTITION BY range (MEASR_COMP_ID) SUBPARTITION TEMPLATE(
    subpartition S01 values less than (124999999999),
    subpartition S02 values less than (249999999999),
    subpartition S03 values less than (374999999999),
    subpartition S04 values less than (499999999999),
    subpartition S05 values less than (624999999999),
    subpartition S06 values less than (744999999999),
    subpartition S07 values less than (874999999999),
    subpartition SMAX values less than (maxvalue)
)
TABLESPACE CM_D1T298_P2011JAN;
TABLESPACE CM_D1T298_P2011MAR;
TABLESPACE CM_D1T298_P2011MAY;
TABLESPACE CM_D1T298_P2011JUL;
TABLESPACE CM_D1T298_P2011SEP;
TABLESPACE CM_D1T298_P2011NOV;
TABLESPACE CM_D1T298_PMAX;

CREATE UNIQUE INDEX D1T298P0 ON D1_MSRMT(MEASR_COMP_ID, MSRMT_DTTM) LOCAL COMPRESS ADVANCED LOW;
ALTER TABLE D1_MSRMT ADD CONSTRAINT D1T298F0 PRIMARY KEY(MEASR_COMP_ID, MSRMT_DTTM) USING INDEX;

CREATE BIGFILE TABLESPACE CM_D1T299_P2011JAN DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T299_P2011MAR DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T299_P2011MAY DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T299_P2011JUL DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T299_P2011SEP DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T299_P2011NOV DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T299_PMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE TABLE D1_MSRMT_CHAR (}
MEASR_COMP_ID CHAR(12) NOT NULL ENABLE, MSRMT_DTTM DATE NOT NULL ENABLE,
CHAR_TYPE_CD   CHAR(8) NOT NULL ENABLE, SEQ_NUM        NUMBER(3,0) NOT NULL ENABLE,
CHAR_VAL       CHAR(16) DEFAULT ' ' NOT NULL ENABLE, ADHOC_CHAR_VAL VARCHAR2(254) DEFAULT ' '
NOT NULL ENABLE, CHAR_VAL_FK1  VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE, CHAR_VAL_FK2
VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE, CHAR_VAL_FK3  VARCHAR2(50) DEFAULT ' ' NOT NULL
ENABLE, CHAR_VAL_FK4  VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE, CHAR_VAL_FK5
VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE, SRCH_CHAR_VAL VARCHAR2(50) DEFAULT ' ' NOT NULL
ENABLE, VERSION        NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
LAST_UPDATE_DTTM DATE, READING_VAL NUMBER(16,6), COMBINED_MULTIPLIER NUMBER(12,6),
READING_COND_FLG CHAR(6)
)
ENABLE ROW MOVEMENT
PARTITION BY RANGE (MSRMT_DTTM) SUBPARTITION BY range (MEASR_COMP_ID) SUBPARTITION
TEMPLATE(
  subpartition S01 values less than (124999999999),
  subpartition S02 values less than (249999999999),
  subpartition S03 values less than (374999999999),
  subpartition S04 values less than (499999999999),
  subpartition S05 values less than (624999999999),
  subpartition S06 values less than (744999999999),
  subpartition S07 values less than (874999999999),
  subpartition SMAX values less than (maxvalue)
)
{
  PARTITION "P2011JAN" VALUES LESS THAN (TO_DATE('2011-02-01 00:00:01', 'SYYYY-MM-DD
                   HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  TABLESPACE CM_D1T299_P2011JAN,
  PARTITION "P2011MAR" VALUES LESS THAN (TO_DATE('2011-04-01 00:00:01', 'SYYYY-MM-DD
                   HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  TABLESPACE CM_D1T299_P2011MAR,
  PARTITION "P2011MAY" VALUES LESS THAN (TO_DATE('2011-06-01 00:00:01', 'SYYYY-MM-DD
                   HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  TABLESPACE CM_D1T299_P2011MAY,
  PARTITION "P2011JUL" VALUES LESS THAN (TO_DATE('2011-08-01 00:00:01', 'SYYYY-MM-DD
                   HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  TABLESPACE CM_D1T299_P2011JUL,
  PARTITION "P2011SEP" VALUES LESS THAN (TO_DATE('2011-10-01 00:00:01', 'SYYYY-MM-DD
                   HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  TABLESPACE CM_D1T299_P2011SEP,
  PARTITION "P2011NOV" VALUES LESS THAN (TO_DATE('2011-12-01 00:00:01', 'SYYYY-MM-DD
                   HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  TABLESPACE CM_D1T299_P2011NOV,
  PARTITION "PMAX" VALUES LESS THAN (MAXVALUE)
  TABLESPACE CM_D1T299_PMAX
); CREATE BIGFILE TABLESPACE CM_D1T299_IND DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE
UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE UNIQUE INDEX D1T299P0 ON D1_MSRMT_CHAR (
MEASR_COMP_ID, MSRMT_DTTM, CHAR_TYPE_CD, SEQ_NUM)
LOCAL COMPRESS ADVANCED LOW;
ALTER TABLE D1_MSRMT_CHAR ADD constraint D1T299P0 PRIMARY KEY (MEASR_COMP_ID, MSRMT_DTTM,
CHAR_TYPE_CD, SEQ_NUM) USING INDEX ;
CREATE INDEX D1T299S1 ON D1_MSRMT_CHAR(SRCH_CHAR_VAL)
GLOBAL PARTITION BY HASH(SRCH_CHAR_VAL)
{
PARTITION P1 TABLESPACE CM_D1T299_IND,
PARTITION P2 TABLESPACE CM_D1T299_IND,
PARTITION P3 TABLESPACE CM_D1T299_IND,
PARTITION P4 TABLESPACE CM_D1T299_IND,
PARTITION P5 TABLESPACE CM_D1T299_IND,
PARTITION P6 TABLESPACE CM_D1T299_IND,
PARTITION P7 TABLESPACE CM_D1T299_IND,
PARTITION P8 TABLESPACE CM_D1T299_IND
)
TABLESPACE CM_D1T304_IND;

D1_MSRMT_LOG
CREATE BIGFILE TABLESPACE CM_D1T300_P2011JAN DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE
UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T300_P2011MAR DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE
UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T300_P2011MAY DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE
UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T300_P2011JUL DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE
UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T300_P2011SEP DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T300_P2011NOV DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T300_PMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;

CREATE TABLE D1_MSRMT_LOG (MEASR_COMP_ID CHAR(12), MSRMT_DTTM DATE, SEQNO NUMBER(5,0), ORIG_INIT_MSRMT_ID CHAR(14) DEFAULT ' ' NOT NULL ENABLE, BUS_OBJ_CD         CHAR(30) DEFAULT ' ' NOT NULL ENABLE, CHAR_TYPE_CD   CHAR(8) DEFAULT ' ' NOT NULL ENABLE, CHAR_VAL       CHAR(16) DEFAULT ' ' NOT NULL ENABLE, ADHOC_CHAR_VAL VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE, CHAR_VAL_FK1 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE, CHAR_VAL_FK2 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE, CHAR_VAL_FK3 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE, CHAR_VAL_FK4 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE, CHAR_VAL_FK5 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE, DESCRLONG VARCHAR2(4000) 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) DEFAULT ' ' NOT NULL ENABLE, VERSION                            NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE, MSRMT_LOG_ENTRY_TYPE_FLG CHAR(4) DEFAULT ' ' NOT NULL ENABLE, BO_DATA_AREA CLOB )
LOB (BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE) ENABLE ROW MOVEMENT PARTITION BY RANGE (MSRMT_DTTM) SUBPARTITION BY range (MEASR_COMP_ID) SUBPARTITION TEMPLATE(
subpartition S01 values less than (124999999999),
subpartition S02 values less than (249999999999),
subpartition S03 values less than (374999999999),
subpartition S04 values less than (499999999999),
subpartition S05 values less than (624999999999),
subpartition S06 values less than (744999999999),
subpartition S07 values less than (874999999999),
subpartition SMAX values less than (maxvalue)
)

PARTITION "P2011JAN" VALUES LESS THAN (TO_DATE('2011-02-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB (BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE) TABLESPACE CM_D1T300_P2011JAN
TABLESPACE CM_D1T300_P2011JAN,
PARTITION "P2011MAR" VALUES LESS THAN (TO_DATE('2011-04-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB (BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE) TABLESPACE CM_D1T300_P2011MAR
TABLESPACE CM_D1T300_P2011MAR,
PARTITION "P2011MAY" VALUES LESS THAN (TO_DATE('2011-06-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB (BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE) TABLESPACE CM_D1T300_P2011MAY
TABLESPACE CM_D1T300_P2011MAY,
PARTITION "P2011JUL" VALUES LESS THAN (TO_DATE('2011-08-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB (BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE) TABLESPACE CM_D1T300_P2011JUL
TABLESPACE CM_D1T300_P2011JUL,
PARTITION "P2011SEP" VALUES LESS THAN (TO_DATE('2011-10-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB (BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE) TABLESPACE CM_D1T300_P2011SEP
TABLESPACE CM_D1T300_P2011SEP,
PARTITION "P2011OCT" VALUES LESS THAN (TO_DATE('2011-11-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB (BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE) TABLESPACE CM_D1T300_P2011OCT
TABLESPACE CM_D1T300_P2011OCT,
PARTITION "PMAX" VALUES LESS THAN (MAXVALUE)
LOB (BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE) TABLESPACE CM_D1T300_PMAX
TABLESPACE CM_D1T300_PMAX);

CREATE UNIQUE INDEX DT1T300P0 ON D1_MSRMT_LOG (MEASR_COMP_ID, MSRMT_DTTM, SEQNO ) LOCAL COMPRESS ADVANCED LOW;
ALTER TABLE D1_MSRMT_LOG ADD CONSTRAINT DT1T300P0 PRIMARY KEY (MEASR_COMP_ID, MSRMT_DTTM, SEQNO) USING INDEX ;
D1_MSRMT_LOG_PARM

CREATE BIGFILE TABLESPACE CM_D1T301_P2011JAN DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T301_P2011MAR DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T301_P2011MAY DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T301_P2011JUL DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T301_P2011SEP DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T301_P2011NOV DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T301_PM1X DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;

CREATE TABLE D1_MSRMT_LOG_PARM (MEASR_COMP_ID CHAR(12), MSRMT_DTTM DATE, SEQNO NUMBER(5,0), PARM_SEQ NUMBER(3,0), MSG_PARM_VAL VARCHAR2(30) DEFAULT ' ' NOT NULL ENABLE, MSG_PARM_TYP_FLG CHAR(4) DEFAULT ' ' NOT NULL ENABLE, VERSION NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE) ENABLE ROW MOVEMENT;

PARTITION BY RANGE (MSRMT_DTTM) SUBPARTITION BY range (MEASR_COMP_ID) SUBPARTITION TEMPLATE (subpartition S01 values less than (124999999999), subpartition S02 values less than (249999999999), subpartition S03 values less than (374999999999), subpartition S04 values less than (499999999999), subpartition S05 values less than (624999999999), subpartition S06 values less than (744999999999), subpartition S07 values less than (874999999999), subpartition SMAX values less than (maxvalue))

CREATE UNIQUE INDEX D1T301P0 ON D1_MSRMT_LOG_PARM (MEASR_COMP_ID, MSRMT_DTTM, SEQNO, PARM_SEQ) INDEX LOCAL COMPRESS ADVANCED LOW;
ALTER TABLE D1_MSRMT_LOG_PARM ADD CONSTRAINT D1T301P0 PRIMARY KEY (MEASR_COMP_ID, MSRMT_DTTM, SEQNO, PARM_SEQ) USING INDEX;

D1_INIT_MSRMT_DATA

CREATE BIGFILE TABLESPACE CM_D1T304_P2011JAN DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011MAR DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011MAY DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011JUL DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011SEP DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T304_P2011NOV DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE BIGFILE TABLESPACE CM_D1T304_PMAX DATAFILE '+DATADG' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED;

CREATE TABLE D1_INIT_MSRMT_DATA
(
    INIT_MSRMT_DATA_ID CHAR(14) NOT NULL ENABLE,
    MEASR_COMP_ID   CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
    D1_FROM_DTTM DATE,
    D1_TO_DTTM DATE,
    DATA_SRC_FLG        CHAR(4) DEFAULT ' ' NOT NULL ENABLE,
    TIME_ZONE_CD        CHAR(10) DEFAULT ' ' NOT NULL ENABLE,
    BUS_GBJ_CD      CHAR(30) DEFAULT ' ' NOT NULL ENABLE,
    BO_STATUS_CD        CHAR(12) DEFAULT ' ' NOT NULL ENABLE,
    BO_STATUS_REASON_CD VARCHAR2(30) DEFAULT ' ' NOT NULL ENABLE,
    IMD_BO_DATA_AREA CLOB,
    STATUS_UPD_DTTM DATE NOT NULL ENABLE,
    CRE_DTTM DATE NOT NULL ENABLE,
    VERSION    NUMBER(5,0) DEFAULT 1 NOT NULL ENABLE,
    IMD_EXT_ID VARCHAR2(120),
    PREVEE_BO_DATA_AREA CLOB,
    POSTVEE_BO_DATA_AREA CLOB,
    TRACE_BO_DATA_AREA CLOB,
    RAW_BO_DATA_AREA CLOB,
    LAST_UPDATE_DTTM DATE,
    ILM_DT DATE,
    ILM_ARCH_SW CHAR(1),
    RETENTION_PERIOD NUMBER(5,0) DEFAULT 99999 NOT NULL ENABLE
)
ENABLE ROW MOVEMENT
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (ENABLE STORAGE IN ROW COMPRESS MEDIUM CACHE)
PARTITION BY RANGE (D1_TO_DTTM)
SUBPARTITION BY range (MEASR_COMP_ID)
SUBPARTITION TEMPLATE(
    SUBPARTITION S01 VALUES LESS THAN (124999999999),
    SUBPARTITION S02 VALUES LESS THAN (249999999999),
    SUBPARTITION S03 VALUES LESS THAN (374999999999),
    SUBPARTITION S04 VALUES LESS THAN (499999999999),
    SUBPARTITION S05 VALUES LESS THAN (624999999999),
    SUBPARTITION S06 VALUES LESS THAN (749999999999),
    SUBPARTITION S07 VALUES LESS THAN (874999999999),
    SUBPARTITION SMAX VALUES LESS THAN (MAXVALUE)
)

PARTITION "P2011JAN" VALUES LESS THAN (TO_DATE('2011-02-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JAN)
TABLESPACE CM_D1T304_P2011JAN,
PARTITION "P2011MAR" VALUES LESS THAN (TO_DATE('2011-04-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAR)
TABLESPACE CM_D1T304_P2011MAR,
PARTITION "P2011MAY" VALUES LESS THAN (TO_DATE('2011-06-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011MAY)
TABLESPACE CM_D1T304_P2011MAY,
PARTITION "P2011JUL" VALUES LESS THAN (TO_DATE('2011-08-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011JUL)
TABLESPACE CM_D1T304_P2011JUL,
PARTITION "P2011SEP" VALUES LESS THAN (TO_DATE('2011-10-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011SEP)
TABLESPACE CM_D1T304_P2011NOV,
PARTITION "P2011NOV" VALUES LESS THAN (TO_DATE('2011-12-01 00:00:01', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GEORGIAN'))
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_P2011NOV)
TABLESPACE CM_D1T304_P2011NOV,
PARTITION "PMAX" VALUES LESS THAN (MAXVALUE)
LOB (PREVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX)
LOB (POSTVEE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX)
LOB (TRACE_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX)
LOB (RAW_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX)
LOB (IMD_BO_DATA_AREA) STORE AS SECUREFILE (TABLESPACE CM_D1T304_PMAX)
TABLESPACE CM_D1T304_PMAX);
CREATE BIGFILE TABLESPACE CM_D1T304_IND DATAFILE '+DATA' SIZE 50M AUTOEXTEND ON MAXSIZE UNLIMITED DEFAULT ROW STORE COMPRESS ADVANCED;
CREATE UNIQUE INDEX D1T304P0 ON D1_INIT_MSRMT_DATA (
INIT_MSRMT_DATA_ID TABLESPACE CM_D1T304_IND
) GLOBAL PARTITION BY RANGE (INIT_MSRMT_DATA_ID)
(PARTITION P1 values less than (12499999999999999),
PARTITION P2 values less than (24999999999999999),
PARTITION P3 values less than (37499999999999999),
PARTITION P4 values less than (49999999999999999),
PARTITION P5 values less than (62499999999999999),
PARTITION P6 values less than (74499999999999999),
PARTITION P7 values less than (87499999999999999),
PARTITION P8 values less than (MAXVALUE));
ALTER TABLE D1_INIT_MSRMT_DATA ADD CONSTRAINT D1T304P0 PRIMARY KEY (INIT_MSRMT_DATA_ID) USING INDEX ;
CREATE INDEX D1T304S1 ON D1_INIT_MSRMT_DATA (MEASR_COMP_ID, BO_STATUS_CD, BUS_OBJ_CD, D1_TO_DTTM, D1_FROM_DTTM) TABLESPACE CM_D1T304_IND
GLOBAL PARTITION BY RANGE (MEASR_COMP_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 ( '749999999999' ),
PARTITION P7 VALUES LESS THAN ( '874999999999' ),
PARTITION P8 VALUES LESS THAN (MAXVALUE));
CREATE UNIQUE INDEX D1T304S3 ON D1_INIT_MSRMT_DATA(IMD_EXT_ID,INIT_MSRMT_DATA_ID) TABLESPACE CM_D1T304_IND
GLOBAL PARTITION BY HASH(IMD_EXT_ID)
(PARTITION P1 TABLESPACE CM_D1T304_IND,
PARTITION P2 TABLESPACE CM_D1T304_IND,
PARTITION P3 TABLESPACE CM_D1T304_IND,
PARTITION P4 TABLESPACE CM_D1T304_IND,
PARTITION P5 TABLESPACE CM_D1T304_IND,
PARTITION P6 TABLESPACE CM_D1T304_IND,
PARTITION P7 TABLESPACE CM_D1T304_IND,
PARTITION P8 TABLESPACE CM_D1T304_IND
) COMPRESS ADVANCED LOW;

D1_INIT_MSRMT_DATA_CHAR
CREATE TABLE D1_INIT_MSRMT_DATA_CHAR
(
INIT_MSRMT_DATA_ID CHAR(14) NOT NULL ENABLE,
CHAR_TYPE_CD CHAR(8) NOT NULL ENABLE,
SEQ_NUM NUMBER(3,0) NOT NULL ENABLE,
CHAR_VAL CHAR(16) DEFAULT ' ' NOT NULL ENABLE,
ADHOC_CHAR_VAL VARCHAR2(254) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK1 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK2 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK3 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,
CHAR_VAL_FK4 VARCHAR2(50) DEFAULT ' ' NOT NULL ENABLE,

Partitioning and Compression Recommendations J - 8
Oracle Utilities Customer To Meter Database Administrator's Guide
### D1_INIT_MSRMT_DATA_K

Create table `D1_INIT_MSRMT_DATA_K`:

```sql
CREATE TABLE D1_INIT_MSRMT_DATA_K (INIT_MSRMT_DATA_ID CHAR(14), ENV_ID NUMBER(6,0) NOT NULL ENABLE, CONSTRAINT D1T314P0 PRIMARY KEY (INIT_MSRMT_DATA_ID, ENV_ID) ENABLE ORGANIZATION INDEX ENABLE ROW MOVEMENT PARTITION BY RANGE (INIT_MSRMT_DATA_ID) (PARTITION P1 values less than (12499999999999), PARTITION P2 values less than (24999999999999), PARTITION P3 values less than (37499999999999), PARTITION P4 values less than (49999999999999), PARTITION P5 values less than (62499999999999), PARTITION P6 values less than (74999999999999), PARTITION P7 values less than (87499999999999), PARTITION P8 values less than (maxvalue)) TABLESPACE CM_D1T314_IND;
```

### D1_INIT_MSRMT_DATA_LOG

Create table `D1_INIT_MSRMT_DATA_LOG`:

```sql
CREATE TABLE D1_INIT_MSRMT_DATA_LOG (INIT_MSRMT_DATA_ID CHAR(14) NOT NULL ENABLE, SEQNO NUMBER(5,0) NOT NULL ENABLE, BO_STATUS_CD CHAR(12) DEFAULT ' ' NOT NULL ENABLE, BO_STATUS_REASON_CD VARCHAR2(30 BYTE) DEFAULT ' ' NOT NULL ENABLE, CHAR_TYPE_CD CHAR(8) DEFAULT ' ' NOT NULL ENABLE, CHAR_VAL CHAR(16) 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) DEFAULT ' ' NOT NULL ENABLE, LOG_DTTM DATE NOT NULL ENABLE,
```

---

**Partitioning and Compression Recommendations**

- **D1_INIT_MSRMT_DATA_K**
  - Create unique index `D1T305P0` on `D1_INIT_MSRMT_DATA_CHAR(INIT_MSRMT_DATA_ID, CHAR_TYPE_CD, SEQ_NUM)` TABLESPACE `CM_D1T304_IND` GLOBAL PARTITION BY RANGE(INIT_MSRMT_DATA_ID)
    - PARTITION P1 VALUES LESS THAN ('12499999999999'), PARTITION P2 VALUES LESS THAN ('24999999999999'), PARTITION P3 VALUES LESS THAN ('37499999999999'), PARTITION P4 VALUES LESS THAN ('49999999999999'), PARTITION P5 VALUES LESS THAN ('62499999999999'), PARTITION P6 VALUES LESS THAN ('74999999999999'), PARTITION P7 VALUES LESS THAN ('87499999999999'), PARTITION P8 VALUES LESS THAN (MAXVALUE)
  - Compress advanced low.

- **D1_INIT_MSRMT_DATA_LOG**
  - Create index `D1T305S1` on `D1_INIT_MSRMT_DATA_CHAR(SRCH_CHAR_VAL)` GLOBAL PARTITION BY HASH(SRCH_CHAR_VAL)
    - PARTITION P1 TABLESPACE CM_D1T304_IND,
    - PARTITION P2 TABLESPACE CM_D1T304_IND,
    - PARTITION P3 TABLESPACE CM_D1T304_IND,
    - PARTITION P4 TABLESPACE CM_D1T304_IND,
    - PARTITION P5 TABLESPACE CM_D1T304_IND,
    - PARTITION P6 TABLESPACE CM_D1T304_IND,
    - PARTITION P7 TABLESPACE CM_D1T304_IND,
    - PARTITION P8 TABLESPACE CM_D1T304_IND

---

Oracle Utilities Customer To Meter Database Administrator's Guide
Compression Recommendations

It is highly recommended to use the following guidelines with regard to compression.

1. For all transactional data tables including ILM enabled tables (except D1_MSRMT* tables):
   a. For easier operational manageability, it is recommended to enable the compression at tablespace level while creating separate tablespaces for each
logical unit of archival (like a parent table partition and the corresponding referenced child table partitions).

b. Use securefile medium compression for LOBs.

c. On Oracle database 12c:
   • Use advanced compression for table data compression.
   • Compress indexes using advanced low compression (using ‘compress advanced low’ clause).

d. On Oracle database 11g:
   • Use OLTP compression for table data and compression indexes using default compression.

2. For D1_MSRMT* tables:
   a. Keep current table partitions uncompressed for D1_MSRMT. Other D1_MSRMT* tables should use compressed tablespaces for all partitions.

   b. For the D1_MSRMT table- Periodically (recommended monthly), compress the data by reloading into a staging table followed by partition exchange. It is highly recommended to use bulk load CTAS operation with parallel clause during the reload.
      • Use ‘QUERY HIGH’ compression for Exadata implementations.
      • For non-Exadata implementations, on 12c use ‘row store compress advanced’ and on 11g use OLTP compression.

   c. For indexes
      • On Oracle database 12c, Compress indexes using advanced low compression (using ‘compress advanced low’ clause).
      • On Oracle database 11g, use default index compression.
Appendix K

Upgrades to the Oracle Utilities Customer To Meter 2.6.0.1.0 Database

This appendix describes database upgrades for the Oracle Utilities Customer To Meter V2.6.0.1.0. 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 V2.6.0.1.0 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 section of the document. The tasks that need to be performed after running the upgrade scripts are included.

The added functionality of V2.6.0.1.0 is not the scope of this documentation. The upgrade scripts do not turn on the newly added functionality by default. For new functionality, refer the V2.6.0.1.0 User Guides. In the last section of this document you will find a list of the new tables that are added in this release.

This section includes:

• Oracle Utilities Customer Care and Billing Database Changes

• Oracle Utilities Meter Data Management Database Changes
Oracle Utilities Customer Care and Billing Database Changes

This section describes database upgrades for the Oracle Utilities Customer Care and Billing V2.6.0.1.0. release, including:

- Schema Changes
- New System Data

Schema Changes

Column Format Changes
The following columns are modified in this Oracle Utilities Customer To Meter release.

<table>
<thead>
<tr>
<th>Table_Name</th>
<th>Column_Name</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>CL_BF_VAL</td>
<td>VAL</td>
<td>NUMBER (18,7)</td>
<td>NUMBER (19,8)</td>
</tr>
<tr>
<td>CL_CVAL_TMPL</td>
<td>VAL</td>
<td>NUMBER (18,7)</td>
<td>NUMBER (19,8)</td>
</tr>
<tr>
<td>CL_CVAL_TOU_TMP</td>
<td>VAL</td>
<td>NUMBER (18,7)</td>
<td>NUMBER (19,8)</td>
</tr>
<tr>
<td>CI_RC</td>
<td>VAL</td>
<td>NUMBER (18,7)</td>
<td>NUMBER (19,8)</td>
</tr>
<tr>
<td>CI_TOU_BF_VAL</td>
<td>VAL</td>
<td>NUMBER (18,7)</td>
<td>NUMBER (19,8)</td>
</tr>
<tr>
<td>CI_TOU_CONT_VAL</td>
<td>VAL</td>
<td>NUMBER (18,7)</td>
<td>NUMBER (19,8)</td>
</tr>
<tr>
<td>CI_SA_CONTERM</td>
<td>VAL</td>
<td>NUMBER (18,7)</td>
<td>NUMBER (19,8)</td>
</tr>
</tbody>
</table>

New System Data
This section lists the new system data that are added for business process configuration.

New Tables
There are no new tables are added to Oracle Utilities Customer Care and Billing in this release.

New Columns
The following columns are added to Oracle Utilities Customer Care and Billing in this release.

<table>
<thead>
<tr>
<th>Table_Name</th>
<th>Column_Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_CC_TYPE</td>
<td>CC_PER_FLG</td>
<td>CHAR(4)</td>
</tr>
<tr>
<td>CI_CC_TYPE</td>
<td>CC_ACCT_FLG</td>
<td>CHAR(4)</td>
</tr>
<tr>
<td>CI_CC_TYPE</td>
<td>CC_PREM_FLG</td>
<td>CHAR(4)</td>
</tr>
<tr>
<td>CI_CC_TYPE</td>
<td>ENTITY_REI_FLG</td>
<td>CHAR(4)</td>
</tr>
</tbody>
</table>
New Indexes
The following indexes are added in this release of Oracle Utilities Customer To Meter.

<table>
<thead>
<tr>
<th>Table_Name</th>
<th>Index_Name</th>
<th>Column_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1_NTF_PREF</td>
<td>C1T002S3</td>
<td>C1_CONTACT_ID</td>
</tr>
<tr>
<td>C1_NTF_PREF</td>
<td>C1T002S4</td>
<td>F1_SVC_TASK_ID</td>
</tr>
</tbody>
</table>

New Functions
There are no new functions added to Oracle Utilities Customer To Meter in this release.

New Batch Control Application Services
The following batch controls use the designated application services:

<table>
<thead>
<tr>
<th>Batch Control</th>
<th>Description</th>
<th>New Application Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTVTAPY</td>
<td>Activate auto-pay</td>
<td>C1-APAY</td>
</tr>
<tr>
<td>ADM</td>
<td>Account debt monitor</td>
<td>ADM</td>
</tr>
<tr>
<td>ADM2</td>
<td>Account debt monitor, minimum days review</td>
<td>ADM2</td>
</tr>
<tr>
<td>ANLYZSAR</td>
<td>Analyze SA relationship</td>
<td>ANLYZSAR</td>
</tr>
<tr>
<td>APAYACH</td>
<td>Auto pay extract - ACH</td>
<td>C1-APAY</td>
</tr>
<tr>
<td>APAYCRET</td>
<td>Create autopay on extract date</td>
<td>C1-APAY</td>
</tr>
<tr>
<td>APAYDSFR</td>
<td>Distribute and freeze autopay</td>
<td>C1-APAY</td>
</tr>
<tr>
<td>APDL</td>
<td>Accounts payable download</td>
<td>APDL</td>
</tr>
<tr>
<td>ASSGNBSBN</td>
<td>Assign sequential bill numbers</td>
<td>ASSGNBSBN</td>
</tr>
<tr>
<td>BALAPY</td>
<td>Create autopay tender controls</td>
<td>BALAPY</td>
</tr>
<tr>
<td>BCASSIGN</td>
<td>Assign balance control id to FTs</td>
<td>C1-BCG</td>
</tr>
<tr>
<td>BCGNEW</td>
<td>Create new balance control group</td>
<td>C1-BCG</td>
</tr>
<tr>
<td>BCGSNAP</td>
<td>Create/validate balance control snapshot</td>
<td>C1-BCG</td>
</tr>
<tr>
<td>BCU1</td>
<td>Billable Charge Upload 1 - Validate Staging</td>
<td>C1-BCU</td>
</tr>
<tr>
<td>BCU2</td>
<td>Billable Charge Upload 2 - Populate</td>
<td>C1-BCU</td>
</tr>
<tr>
<td>BILLING</td>
<td>Create bills using bill cycle</td>
<td>BILLING</td>
</tr>
<tr>
<td>BUDMON</td>
<td>Monitor budgets</td>
<td>BUDMON</td>
</tr>
<tr>
<td>BUDTRUP</td>
<td>True up budgets</td>
<td>BUDTRUP</td>
</tr>
<tr>
<td>C1-ACAMT</td>
<td>Update Bill Segment Audit Calc Amount</td>
<td>C1-ACAMT</td>
</tr>
<tr>
<td>C1-ACTRQ</td>
<td>Activity Request Monitor (Deferred)</td>
<td>C1-ACTRQ</td>
</tr>
<tr>
<td>C1-ADAMT</td>
<td>Update Adjustment Audit Calc Amount</td>
<td>C1-ADAMT</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
<td>New Application Service</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>C1-ADMOV</td>
<td>Overdue Monitor</td>
<td>C1-ADMOV</td>
</tr>
<tr>
<td>C1-ADUP1</td>
<td>Adjustment Upload Preprocessor</td>
<td>C1-ADUP</td>
</tr>
<tr>
<td>C1-ADUP2</td>
<td>Adjustment Upload</td>
<td>C1-ADUP</td>
</tr>
<tr>
<td>C1-ADURS</td>
<td>Resolve Suspense Adjustments</td>
<td>C1-ADURS</td>
</tr>
<tr>
<td>C1-APACH</td>
<td>Auto pay extract - ACH (with offset days parameter)</td>
<td>C1-APAY</td>
</tr>
<tr>
<td>C1-APRPR</td>
<td>Approval Request Monitor</td>
<td>C1-APRREQ</td>
</tr>
<tr>
<td>C1-APRTR</td>
<td>Approval Request Monitor (Deferred)</td>
<td>C1-APRREQ</td>
</tr>
<tr>
<td>C1-ARQPR</td>
<td>Activity Request Monitor</td>
<td>C1-ACTREQ</td>
</tr>
<tr>
<td>C1-BILL</td>
<td>Billing</td>
<td>BILLING</td>
</tr>
<tr>
<td>C1-BLAPC</td>
<td>Balance Autopay Control Tables</td>
<td>BALAPY</td>
</tr>
<tr>
<td>C1-BLCMP</td>
<td>Freeze and Complete Pending Bills</td>
<td>C1-BLCMP</td>
</tr>
<tr>
<td>C1-BLEIL</td>
<td>Billing Data Initial Load for DataConnect</td>
<td>C1-DATACON</td>
</tr>
<tr>
<td>C1-BLRV1</td>
<td>Bill Review Insertion</td>
<td>C1-BLRV</td>
</tr>
<tr>
<td>C1-BLRVV</td>
<td>Bill Review Validation</td>
<td>C1-BLRVV</td>
</tr>
<tr>
<td>C1-BLRVW</td>
<td>Review Bills for Settlement of Held GL Amounts</td>
<td>C1-BLRVW</td>
</tr>
<tr>
<td>C1-BNBAS</td>
<td>Assign Bill Document Numbers</td>
<td>C1-DOCNBR</td>
</tr>
<tr>
<td>C1-BSYEX</td>
<td>Billing Data Extract for DataConnect</td>
<td>C1-DATACON</td>
</tr>
<tr>
<td>C1-CPRPR</td>
<td>Conservation Program Monitor Process</td>
<td>C1-CONSRVMON</td>
</tr>
<tr>
<td>C1-CPRQ</td>
<td>Conservation Program Monitor Process (Deferred)</td>
<td>C1-CONSRVMON</td>
</tr>
<tr>
<td>C1-CRRMP</td>
<td>Customer Relationship Request Periodic Monitor Process</td>
<td>C1-CRRMP</td>
</tr>
<tr>
<td>C1-CSTRS</td>
<td>Case Scheduled Transition</td>
<td>C1-CASETRAN</td>
</tr>
<tr>
<td>C1-CTCOE</td>
<td>Complete To Do Entry of Contract Option Event Exception</td>
<td>C1-CTCOE</td>
</tr>
<tr>
<td>C1-CTIDS</td>
<td>Complete To Do Entry of Interval Data Set Exception</td>
<td>C1-CTIDS</td>
</tr>
<tr>
<td>C1-CTRDS</td>
<td>Complete To Do Entry of Interval Register Data Set Exception</td>
<td>C1-CTRDS</td>
</tr>
<tr>
<td>C1-CTTDS</td>
<td>Complete To Do Entry of TOU Data Set Exception</td>
<td>C1-CTTDS</td>
</tr>
<tr>
<td>C1-DVTSC</td>
<td>Cancel Device Test Selection</td>
<td>C1-DVTSC</td>
</tr>
<tr>
<td>C1-ECRVL</td>
<td>Encrypt Legacy Order Field Col Ref Value</td>
<td>C1-ENCVR</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
<td>New Application Service</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>C1-INITR</td>
<td>Initiative Periodic Monitor Process</td>
<td>C1-INITIATIVE</td>
</tr>
<tr>
<td>C1-INPUS</td>
<td>Create Person Contact from Person Phone/ Email</td>
<td>C1-INPUS</td>
</tr>
<tr>
<td>C1-ISSCT</td>
<td>Issuing Center Monitor</td>
<td>C1-ISSCT</td>
</tr>
<tr>
<td>C1-LDEVT</td>
<td>Lead Event Periodic Monitor Process</td>
<td>C1-INITIATIVE</td>
</tr>
<tr>
<td>C1-LDEXT</td>
<td>Advanced Analysis System Lead - Extract</td>
<td>C1-LEAD</td>
</tr>
<tr>
<td>C1-LDRTY</td>
<td>Advanced Analysis System Lead - Retry</td>
<td>C1-LEAD</td>
</tr>
<tr>
<td>C1-LDTR</td>
<td>Lead Periodic Monitor Process</td>
<td>C1-INITIATIVE</td>
</tr>
<tr>
<td>C1-LEADD</td>
<td>Lead Disposition</td>
<td>C1-INITIATIVBOAS</td>
</tr>
<tr>
<td>C1-LEADG</td>
<td>Lead Generation</td>
<td>C1-INITIATIVBOAS</td>
</tr>
<tr>
<td>C1-MMTPR</td>
<td>Market Message Type Monitor</td>
<td>C1-MMTPR</td>
</tr>
<tr>
<td>C1-NEMDF</td>
<td>NEM Scheduled Monitor Process (Deferred)</td>
<td>C1-NEMDF</td>
</tr>
<tr>
<td>C1-OCBG</td>
<td>Off Cycle Bill Generator Monitor</td>
<td>C1-OCBG</td>
</tr>
<tr>
<td>C1-ODET</td>
<td>Overdue and Cut Event Manager</td>
<td>C1-ODET</td>
</tr>
<tr>
<td>C1-PAYTP</td>
<td>Payment Template Monitor</td>
<td>C1-PAYTP</td>
</tr>
<tr>
<td>C1-PEPL1</td>
<td>Payment Event Upload Stage 1</td>
<td>C1-PEPL</td>
</tr>
<tr>
<td>C1-PEPL2</td>
<td>Payment Event Upload Stage 2</td>
<td>C1-PEPL</td>
</tr>
<tr>
<td>C1-PEPL3</td>
<td>Balance Tender Controls</td>
<td>C1-PEPL</td>
</tr>
<tr>
<td>C1-PNBAS</td>
<td>Assign Payment Event Document Numbers</td>
<td>C1-DOCNBR</td>
</tr>
<tr>
<td>C1-PPBER</td>
<td>Prepay Biller Task - Error</td>
<td>C1-PPB</td>
</tr>
<tr>
<td>C1-PPBTR</td>
<td>Prepay Biller Task</td>
<td>C1-PPB</td>
</tr>
<tr>
<td>C1-PUBAL</td>
<td>Payment upload balance control tables</td>
<td>C1-PUPL</td>
</tr>
<tr>
<td>C1-RCTRQ</td>
<td>Rebate Claim Monitor</td>
<td>C1-RCTRQ</td>
</tr>
<tr>
<td>C1-RDFPR</td>
<td>Conservation Program Rebate Definition Monitor</td>
<td>C1-CONSRVMON</td>
</tr>
<tr>
<td>C1-SAEIL</td>
<td>SA-Based Initial Load for DataConnect</td>
<td>C1-DATACON</td>
</tr>
<tr>
<td>C1-SAFTCT</td>
<td>SAFT-PT Audit Extract Concatenator</td>
<td>C1-SAFTPT</td>
</tr>
<tr>
<td>C1-SAFTPT</td>
<td>SAFT-PT Audit Extract</td>
<td>C1-SAFTPT</td>
</tr>
<tr>
<td>C1-SASYX</td>
<td>SA-Based Extract for DataConnect</td>
<td>C1-DATACON</td>
</tr>
<tr>
<td>C1-SDDCE</td>
<td>SEPA Direct Debit Payment Extract</td>
<td>C1-SDDCE</td>
</tr>
<tr>
<td>C1-SMEIL</td>
<td>Meter History Initial Load for DataConnect</td>
<td>C1-DATACON</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
<td>New Application Service</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>C1-SMISL</td>
<td>Synchronize Meter/Item Stock Location</td>
<td>C1-SMISL</td>
</tr>
<tr>
<td>C1-SMSYX</td>
<td>Meter History Extract for DataConnect</td>
<td>C1-DATACON</td>
</tr>
<tr>
<td>C1-SPEIL</td>
<td>SP-Based Initial Load for DataConnect</td>
<td>C1-DATACON</td>
</tr>
<tr>
<td>C1-SPSYX</td>
<td>SP-Based Extract for DataConnect</td>
<td>C1-DATACON</td>
</tr>
<tr>
<td>C1-SRDS</td>
<td>Service Route Download Staging</td>
<td>C1-SRDS</td>
</tr>
<tr>
<td>C1-SYNIL</td>
<td>Generic Sync Request Initial Load</td>
<td>C1-SYNIL</td>
</tr>
<tr>
<td>C1-TCRNB</td>
<td>Transition OCBG for Replacement Read Process</td>
<td>C1-TCRNB</td>
</tr>
<tr>
<td>C1-TDCOE</td>
<td>Create To Do Entry for Contract Option Event Exception</td>
<td>C1-TDCOE</td>
</tr>
<tr>
<td>C1-TDIDS</td>
<td>Create To Do Entry for Interval Data Set Exception</td>
<td>C1-TDIDS</td>
</tr>
<tr>
<td>C1-TDRDS</td>
<td>Create To Do Entry for Interval Register Data Set Exception</td>
<td>C1-TDRDS</td>
</tr>
<tr>
<td>C1-TDTDS</td>
<td>Create To Do Entry for TOU Data Set Exception</td>
<td>C1-TDTDS</td>
</tr>
<tr>
<td>C1-TOUTR</td>
<td>TOU Map Data Generation Monitor</td>
<td>C1-TOUTR</td>
</tr>
<tr>
<td>C1-UPDBF</td>
<td>Update Calc Rules Mapping</td>
<td>C1-UPDBF</td>
</tr>
<tr>
<td>C1-UPDNT</td>
<td>Update Notification Tasks</td>
<td>C1-UPDNT</td>
</tr>
<tr>
<td>C1-UPDPT</td>
<td>Update Payment Tender Alternate Currency</td>
<td>C1-UPDPT</td>
</tr>
<tr>
<td>C1-USGDF</td>
<td>Usage Scheduled Monitor Process (Deferred)</td>
<td>C1-USAGE</td>
</tr>
<tr>
<td>C1-USGTR</td>
<td>Usage Periodic Monitor Process</td>
<td>C1-USAGE</td>
</tr>
<tr>
<td>C1-WAMAS</td>
<td>Extract asset data for work and asset management</td>
<td>C1-WAMEXTRACT</td>
</tr>
<tr>
<td>C1-WAMEX</td>
<td>Extract customer data for work and asset management</td>
<td>C1-WAMEXTRACT</td>
</tr>
<tr>
<td>C1-WFSUB</td>
<td>Workflow event trigger - Batch scheduler</td>
<td>C1-WORKFLOW</td>
</tr>
<tr>
<td>CAREPROG</td>
<td>Create customer contact for expiring SA char</td>
<td>CAREPROG</td>
</tr>
<tr>
<td>CASETRAN</td>
<td>Case Status Automatic Transition Process</td>
<td>C1-CASETRAN</td>
</tr>
<tr>
<td>CET</td>
<td>Collection event trigger</td>
<td>CET</td>
</tr>
<tr>
<td>CLOSEQTE</td>
<td>Close expired quotes</td>
<td>CLOSEQTE</td>
</tr>
<tr>
<td>CPCRALOC</td>
<td>Capital credit allocation</td>
<td>C1-CAPCRE</td>
</tr>
<tr>
<td>CPCRRRETR</td>
<td>Capital credit retirement</td>
<td>C1-CAPCRE</td>
</tr>
<tr>
<td>CPM</td>
<td>Collection process monitor</td>
<td>CPM</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
<td>New Application Service</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>DEPINTRF</td>
<td>Deposit interest refund</td>
<td>C1-DEPOSIT</td>
</tr>
<tr>
<td>DEPRFND</td>
<td>Deposit refund</td>
<td>C1-DEPOSIT</td>
</tr>
<tr>
<td>DEPRVW</td>
<td>Deposit review</td>
<td>C1-DEPOSIT</td>
</tr>
<tr>
<td>DSGPFODL</td>
<td>Dispatch Group Field Order Print</td>
<td>C1-FODWNLD</td>
</tr>
<tr>
<td>DWLDBILC</td>
<td>Download billable charge</td>
<td>DWLDBILC</td>
</tr>
<tr>
<td>DWLDCOLL</td>
<td>Download collection agency ref</td>
<td>DWLDCOLL</td>
</tr>
<tr>
<td>DWLDCONS</td>
<td>Download consumption</td>
<td>DWLDCONS</td>
</tr>
<tr>
<td>F1-ENCRS</td>
<td>Encrypt Legacy Schema Field Data</td>
<td>C1-ENCR</td>
</tr>
<tr>
<td>F1-ENCRT</td>
<td>Encrypt Legacy Table Field Data</td>
<td>C1-ENCR</td>
</tr>
<tr>
<td>FACOMPL</td>
<td>Field activity completion</td>
<td>C1-FAUPLD</td>
</tr>
<tr>
<td>FACT</td>
<td>Field activity remark activation</td>
<td>C1-FAUPLD</td>
</tr>
<tr>
<td>FANRMRCO</td>
<td>Complete FA using a recent MR</td>
<td>FANRMRCO</td>
</tr>
<tr>
<td>FAXROUT</td>
<td>Fax Routing</td>
<td>FAXROUT</td>
</tr>
<tr>
<td>FDS</td>
<td>Field order download staging</td>
<td>C1-FODWNLD</td>
</tr>
<tr>
<td>FOD</td>
<td>Automatic dispatch of FA's</td>
<td>C1-FODWNLD</td>
</tr>
<tr>
<td>FODL</td>
<td>Field order download extract</td>
<td>C1-FODWNLD</td>
</tr>
<tr>
<td>GLASSIGN</td>
<td>Populates GL_ACCT on CI_FT_GL</td>
<td>C1-GL</td>
</tr>
<tr>
<td>GLDL</td>
<td>GL download extract</td>
<td>C1-GL</td>
</tr>
<tr>
<td>GLS</td>
<td>Create GL download staging</td>
<td>C1-GL</td>
</tr>
<tr>
<td>IB-SPDB</td>
<td>Interval data set derivation</td>
<td>C1-INTERVAL</td>
</tr>
<tr>
<td>IB-STDB</td>
<td>SA specific TOU data creation</td>
<td>C1-INTERVAL</td>
</tr>
<tr>
<td>IPDSDVB</td>
<td>Interval prof data validation</td>
<td>C1-INTERVAL</td>
</tr>
<tr>
<td>IPDSIDB</td>
<td>Determine profile for datasets</td>
<td>C1-INTERVAL</td>
</tr>
<tr>
<td>IREGDVB</td>
<td>Interval reg. data validation</td>
<td>C1-INTERVAL</td>
</tr>
<tr>
<td>IREGIDB</td>
<td>Determine register for datasets</td>
<td>C1-INTERVAL</td>
</tr>
<tr>
<td>LATEPYMT</td>
<td>Create late payment charges</td>
<td>LATEPYMT</td>
</tr>
<tr>
<td>LTRPRT</td>
<td>Letter Extract</td>
<td>LTRPRT</td>
</tr>
<tr>
<td>MASSCNCL</td>
<td>Mass bill cancellation</td>
<td>MASSCNCL</td>
</tr>
<tr>
<td>MASSROBL</td>
<td>Mass re-open of bills</td>
<td>MASSROBL</td>
</tr>
<tr>
<td>MDL</td>
<td>Meter read download extract</td>
<td>MDL</td>
</tr>
<tr>
<td>MDS</td>
<td>Create MR download staging</td>
<td>C1-MRDWNLD</td>
</tr>
<tr>
<td>MRRA</td>
<td>Execute MR remark algorithms</td>
<td>MRRA</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
<td>New Application Service</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>MSR</td>
<td>Create MR schedule routes</td>
<td>C1-MRDWNLD</td>
</tr>
<tr>
<td>MUP1</td>
<td>Meter Read upload 1 - populate meter config</td>
<td>C1-MRUPLD</td>
</tr>
<tr>
<td>MUP2</td>
<td>Meter Read upload 2 - populate meter read</td>
<td>C1-MRUPLD</td>
</tr>
<tr>
<td>NBBAPAY</td>
<td>Create autopay for NBB's</td>
<td>C1-NBB</td>
</tr>
<tr>
<td>NBBPS</td>
<td>Process NBB scheduled payments</td>
<td>C1-NBB</td>
</tr>
<tr>
<td>NDSXTR</td>
<td>Notif download staging extract</td>
<td>NDSXTR</td>
</tr>
<tr>
<td>PAYSPR</td>
<td>Pay service provider</td>
<td>PAYSPR</td>
</tr>
<tr>
<td>POSTROUT</td>
<td>Postal bill routing</td>
<td>C1-BILLPRINT</td>
</tr>
<tr>
<td>PPAPAY</td>
<td>Generate autopay for pay plans</td>
<td>C1-PAYPLAN</td>
</tr>
<tr>
<td>PPM</td>
<td>Pay plan monitor</td>
<td>C1-PAYPLAN</td>
</tr>
<tr>
<td>PSASPM</td>
<td>Pending SA/SP monitor</td>
<td>PSASPM</td>
</tr>
<tr>
<td>PUPL</td>
<td>Payment upload</td>
<td>C1-PUPL</td>
</tr>
<tr>
<td>PY-RPE</td>
<td>Resolve payments in error</td>
<td>PY-RPE</td>
</tr>
<tr>
<td>QUOTROUT</td>
<td>Quote routing</td>
<td>QUOTROUT</td>
</tr>
<tr>
<td>REACH</td>
<td>YTD charitable contributions</td>
<td>REACH</td>
</tr>
<tr>
<td>REDSAAMT</td>
<td>Update old FT's as redundant</td>
<td>REDSAAMT</td>
</tr>
<tr>
<td>REGCNST</td>
<td>Register constant validation</td>
<td>REGCNST</td>
</tr>
<tr>
<td>RTTYPOST</td>
<td>Postal bill routing using bill print software</td>
<td>C1-BILLPRINT</td>
</tr>
<tr>
<td>SAACT</td>
<td>Activate pending start/stop SA</td>
<td>SAACT</td>
</tr>
<tr>
<td>SAEXPIRE</td>
<td>Stop expired SAs</td>
<td>SAEXPIRE</td>
</tr>
<tr>
<td>SARENEW</td>
<td>Renew Service Agreement</td>
<td>SARENEW</td>
</tr>
<tr>
<td>SASP</td>
<td>Find read for SA/SP</td>
<td>SASP</td>
</tr>
<tr>
<td>SEC</td>
<td>Severance event completion</td>
<td>C1-SEVEVT</td>
</tr>
<tr>
<td>SED</td>
<td>Severance event set dependency date</td>
<td>C1-SEVEVT</td>
</tr>
<tr>
<td>SET</td>
<td>Severance event trigger</td>
<td>C1-SEVEVT</td>
</tr>
<tr>
<td>STMDWLD</td>
<td>Download statements</td>
<td>STMDWLD</td>
</tr>
<tr>
<td>STMPRD</td>
<td>Create statements</td>
<td>STMPRD</td>
</tr>
<tr>
<td>TD-BCUPL</td>
<td>To Do for Billable Charge in Error</td>
<td>C1-BILLERR</td>
</tr>
<tr>
<td>TD-BIERR</td>
<td>To Do for Bills in Error</td>
<td>C1-BILLERR</td>
</tr>
<tr>
<td>TD-BSERR</td>
<td>To Do for Bill Segment in Error</td>
<td>C1-BILLERR</td>
</tr>
<tr>
<td>TD-BTERR</td>
<td>To Do for Batch Errors</td>
<td>TD-BTERR</td>
</tr>
<tr>
<td>TD-CCCB</td>
<td>To Do for Customer Contact</td>
<td>TD-CCCB</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
<td>New Application Service</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>TD-CEVT</td>
<td>To Do for C&amp;C Events</td>
<td>TD-CEVT</td>
</tr>
<tr>
<td>TD-DTCS1</td>
<td>Create To Do for Deposit/Tender Upload Error</td>
<td>TD-DTCS1</td>
</tr>
<tr>
<td>TD-DTCS2</td>
<td>Complete To Do for Deposit/Tender Upload Error</td>
<td>TD-DTCS2</td>
</tr>
<tr>
<td>TD-DTCST</td>
<td>To Do for Deposit/Tender Upload Error</td>
<td>TD-DTCST</td>
</tr>
<tr>
<td>TD-ECBK</td>
<td>To Do for Callback Orders</td>
<td>TD-ECBK</td>
</tr>
<tr>
<td>TD-EPND</td>
<td>To Do for Pending Orders</td>
<td>TD-EPND</td>
</tr>
<tr>
<td>TD-FACT</td>
<td>To Do for Field Activity Remark Exception</td>
<td>TD-FACT</td>
</tr>
<tr>
<td>TD-FAUPL</td>
<td>To Do for Field Activity Upload in Error</td>
<td>TD-FAUPL</td>
</tr>
<tr>
<td>TD-HILO</td>
<td>To Do for Meter Read High/Low Errors</td>
<td>TD-HILO</td>
</tr>
<tr>
<td>TD-MODTL</td>
<td>To Do for Open-Dispute Match Event</td>
<td>TD-MODTL</td>
</tr>
<tr>
<td>TD-MONTL</td>
<td>To Do for Open/non-Dispute Match Event</td>
<td>TD-MONTL</td>
</tr>
<tr>
<td>TD-MRRER</td>
<td>To Do for Meter Read Remarks in Error</td>
<td>TD-MRRER</td>
</tr>
<tr>
<td>TD-MRUPL</td>
<td>To Do for Meter Read Upload in Error</td>
<td>TD-MRUPL</td>
</tr>
<tr>
<td>TD-NCDEX</td>
<td>To Do for Non Cash Deposit</td>
<td>TD-NCDEX</td>
</tr>
<tr>
<td>TD-NOBC</td>
<td>To Do for Account without a Bill Cycle</td>
<td>TD-NOBC</td>
</tr>
<tr>
<td>TD-NOMR</td>
<td>To Do for SP without Meter Read Cycle</td>
<td>TD-NOMR</td>
</tr>
<tr>
<td>TD-NTDWN</td>
<td>To Do for NT Download in Error</td>
<td>TD-NTDWN</td>
</tr>
<tr>
<td>TD-NTUPL</td>
<td>To Do for Notification Upload in Error</td>
<td>TD-NTUPL</td>
</tr>
<tr>
<td>TD-PYERR</td>
<td>To Do for Payments in Error/Unfrozen</td>
<td>TD-PYERR</td>
</tr>
<tr>
<td>TD-PYUPL</td>
<td>To Do for Payment Staging Error</td>
<td>TD-PYUPL</td>
</tr>
<tr>
<td>TD-SEVT</td>
<td>To Do for Severance Events</td>
<td>TD-SEVT</td>
</tr>
<tr>
<td>TD-SPRO</td>
<td>To Do for Severance Processes</td>
<td>TD-SPRO</td>
</tr>
<tr>
<td>TD-SSFTL</td>
<td>To Do for Old Pending Start/Stops</td>
<td>TD-SSFTL</td>
</tr>
<tr>
<td>TD-UNBAL</td>
<td>To Do for Unbalanced Pay Event</td>
<td>TD-UNBAL</td>
</tr>
<tr>
<td>TD-WEXTL</td>
<td>To Do for Workflow Events in Error</td>
<td>TD-WEXTL</td>
</tr>
<tr>
<td>TD-WOEVT</td>
<td>To Do for Write-Off Events</td>
<td>TD-WOEVT</td>
</tr>
<tr>
<td>TD-XAIDN</td>
<td>To Do for XAI Download Staging in error</td>
<td>TD-XAIDN</td>
</tr>
<tr>
<td>TD-XAIUP</td>
<td>To Do for XAI Upload Staging in error</td>
<td>TD-XAIUP</td>
</tr>
<tr>
<td>TREND</td>
<td>Trend update</td>
<td>TREND</td>
</tr>
<tr>
<td>UARENEW</td>
<td>Umbrella Agreement Renewal</td>
<td>UARENEW</td>
</tr>
<tr>
<td>UPDERR</td>
<td>Update Batch Run/Thread Status</td>
<td>UPDERR</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
<td>New Application Service</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>WAITCOM</td>
<td>Completes a WF Event in Waiting state</td>
<td>C1-WORKFLOW</td>
</tr>
<tr>
<td>WAITFA</td>
<td>Wait for field actv completion</td>
<td>C1-WORKFLOW</td>
</tr>
<tr>
<td>WAITMAN</td>
<td>Workflow timeout manual wait</td>
<td>C1-WORKFLOW</td>
</tr>
<tr>
<td>WAITNT</td>
<td>Wait for notification response</td>
<td>C1-WORKFLOW</td>
</tr>
<tr>
<td>WET</td>
<td>Write off event trigger</td>
<td>WET</td>
</tr>
<tr>
<td>WFET</td>
<td>Workflow event trigger</td>
<td>C1-WORKFLOW</td>
</tr>
<tr>
<td>WFPRINIT</td>
<td>Workflow process initiation</td>
<td>C1-WORKFLOW</td>
</tr>
<tr>
<td>WPM</td>
<td>Write off monitor process</td>
<td>WPM</td>
</tr>
<tr>
<td>WX-NOTIF</td>
<td>Self-Service Notification Monitor</td>
<td>WX-NOTIF</td>
</tr>
</tbody>
</table>
### Conversion Batch Control Application Services

The following batch controls use the C1-CONVERSION application service:

<table>
<thead>
<tr>
<th>Batch Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIPVAAAPI</td>
<td>Insert CI_ACCT_APAY</td>
</tr>
<tr>
<td>CIPVAAAPK</td>
<td>Generate CI_ACCT_APAY keys</td>
</tr>
<tr>
<td>CIPVAAAPV</td>
<td>Foreign Key validation for CI_ACCT_APAY</td>
</tr>
<tr>
<td>CIPVACCI</td>
<td>Insert CI_ACCT</td>
</tr>
<tr>
<td>CIPVACCK</td>
<td>Generate CI_ACCT keys</td>
</tr>
<tr>
<td>CIPVACHI</td>
<td>Insert CI_ACCT_CHAR</td>
</tr>
<tr>
<td>CIPVACHV</td>
<td>Foreign Key validation for CI_ACCT_CHAR</td>
</tr>
<tr>
<td>CIPVACPI</td>
<td>Insert CI_ACCT_PER</td>
</tr>
<tr>
<td>CIPVACPV</td>
<td>Foreign Key validation for CI_ACCT_PER</td>
</tr>
<tr>
<td>CIPVADCI</td>
<td>Insert CI_ADJ_CHAR</td>
</tr>
<tr>
<td>CIPVADCV</td>
<td>Foreign Key validation for CI_ADJ_CHAR</td>
</tr>
<tr>
<td>CIPVADJ</td>
<td>Insert CI_ADJ</td>
</tr>
<tr>
<td>CIPVADJK</td>
<td>Generate CI_ADJ keys</td>
</tr>
<tr>
<td>CIPVADJV</td>
<td>Foreign Key validation for CI_ADJ</td>
</tr>
<tr>
<td>CIPVAPAI</td>
<td>Insert CI_PRM_ALT_ADDR</td>
</tr>
<tr>
<td>CIPVAPAK</td>
<td>Generate CI_PRM_ALT_ADDR keys</td>
</tr>
<tr>
<td>CIPVAPAV</td>
<td>Foreign Key validation for CI_PRM_ALT_ADDR</td>
</tr>
<tr>
<td>CIPVAPRI</td>
<td>Insert CI_ADJ_APREQ</td>
</tr>
<tr>
<td>CIPVAPRK</td>
<td>Generate CI_ADJ_APREQ keys</td>
</tr>
<tr>
<td>CIPVAPRV</td>
<td>Foreign Key validation for CI_ADJ_APREQ</td>
</tr>
<tr>
<td>CIPVARHI</td>
<td>Insert CI_COLL_AGY_HIS</td>
</tr>
<tr>
<td>CIPVARHV</td>
<td>Foreign Key validation for CI_COLL_AGY_HIS</td>
</tr>
<tr>
<td>CIPVARS</td>
<td>Insert CI_ADM_RVV_SCH</td>
</tr>
<tr>
<td>CIPVARSV</td>
<td>Foreign Key validation for CI_ADM_RVV_SCH</td>
</tr>
<tr>
<td>CIPVBCCI</td>
<td>Insert CI_BSEG_CL_CHAR</td>
</tr>
<tr>
<td>CIPVBCCV</td>
<td>Foreign Key validation for CI_BSEG_CL_CHAR</td>
</tr>
<tr>
<td>CIPVBGI</td>
<td>Insert CI_BILL_CHG</td>
</tr>
<tr>
<td>CIPVBGK</td>
<td>Generate CI_BILL_CHG keys</td>
</tr>
<tr>
<td>CIPVBCGV</td>
<td>Foreign Key validation for CI_BILL_CHG</td>
</tr>
<tr>
<td>CIPVBCHI</td>
<td>Insert CI_BILL_CHAR</td>
</tr>
<tr>
<td>CIPVBCHV</td>
<td>Foreign Key validation for CI_BILL_CHAR</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CIPVBCLI</td>
<td>Insert CI_B_CHG_LINE</td>
</tr>
<tr>
<td>CIPVBCLV</td>
<td>Foreign Key validation for CI_B_CHG_LINE</td>
</tr>
<tr>
<td>CIPVBFFVI</td>
<td>Insert CI_BF_VAL</td>
</tr>
<tr>
<td>CIPVBFFVV</td>
<td>Foreign Key validation for CI_BF_VAL</td>
</tr>
<tr>
<td>CIPVBILLK</td>
<td>Generate CI_BILL keys</td>
</tr>
<tr>
<td>CIPVBILLI</td>
<td>Insert CI_BILL</td>
</tr>
<tr>
<td>CIPVBLLLV</td>
<td>Foreign Key validation for CI_BILL</td>
</tr>
<tr>
<td>CIPVBLLMI</td>
<td>Insert CI_BILL_MSGS</td>
</tr>
<tr>
<td>CIPVBLLMV</td>
<td>Foreign Key validation for CI_BILL_MSGS</td>
</tr>
<tr>
<td>CIPVBLLRI</td>
<td>Insert CI_BILL_ROUTING</td>
</tr>
<tr>
<td>CIPVBLLRV</td>
<td>Foreign Key validation for CI_BILL_ROUTING</td>
</tr>
<tr>
<td>CIPVBRVI</td>
<td>Insert CI_BILL_RVW_SCH</td>
</tr>
<tr>
<td>CIPVBRVV</td>
<td>Foreign Key validation for CI_BILL_RVW_SCH</td>
</tr>
<tr>
<td>CIPVBSAI</td>
<td>Insert CI_BILL_SA</td>
</tr>
<tr>
<td>CIPVBSAV</td>
<td>Foreign Key validation for CI_BILL_SA</td>
</tr>
<tr>
<td>CIPVBSCII</td>
<td>Insert CI_BSEG_CALC</td>
</tr>
<tr>
<td>CIPVBSCIV</td>
<td>Foreign Key validation for CI_BSEG_CALC</td>
</tr>
<tr>
<td>CIPVBSGK</td>
<td>Generate CI_BSEG keys</td>
</tr>
<tr>
<td>CIPVBSSI</td>
<td>Insert CI_BSEG_ITEM</td>
</tr>
<tr>
<td>CIPVBSIV</td>
<td>Foreign Key validation for CI_BSEG_ITEM</td>
</tr>
<tr>
<td>CIPVBSSLI</td>
<td>Insert CI_BSEG_CALC_LN</td>
</tr>
<tr>
<td>CIPVBSSLV</td>
<td>Foreign Key validation for CI_BSEG_CALC_LN</td>
</tr>
<tr>
<td>CIPVCARI</td>
<td>Insert CI_COLL_AGY_REF</td>
</tr>
<tr>
<td>CIPVCARK</td>
<td>Generate CI_COLL_AGY_REF keys</td>
</tr>
<tr>
<td>CIPVCARV</td>
<td>Foreign Key validation for CI_COLL_AGY_REF</td>
</tr>
<tr>
<td>CIPVCCAI</td>
<td>Insert CI_COLL_AGY_REF</td>
</tr>
<tr>
<td>CIPVCCFK</td>
<td>Foreign Key validation for CI_COLL_AGY_REF</td>
</tr>
<tr>
<td>CIPVCCHII</td>
<td>Insert CI_CC_CHAR</td>
</tr>
<tr>
<td>CIPVCCHV</td>
<td>Foreign Key validation for CI_CC_CHAR</td>
</tr>
<tr>
<td>CIPVCCLI</td>
<td>Insert CI_CC_LOG</td>
</tr>
<tr>
<td>CIPVCCLV</td>
<td>Foreign Key validation for CI_CC_LOG</td>
</tr>
<tr>
<td>CIPVCCCTK</td>
<td>Generate CI_CC keys</td>
</tr>
<tr>
<td>CIPVCCECI</td>
<td>Insert CI_COLL_EVT_CC</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CIPVCECV</td>
<td>Foreign Key validation for CI_COLL_EVT_CC</td>
</tr>
<tr>
<td>CIPVCEVI</td>
<td>Insert CI_COLL_EVT</td>
</tr>
<tr>
<td>CIPVCEVK</td>
<td>Generate CI_COLL_EVT keys</td>
</tr>
<tr>
<td>CIPVCLPI</td>
<td>Insert CI_COLL_PROC</td>
</tr>
<tr>
<td>CIPVCLPK</td>
<td>Generate CI_COLL_PROC keys</td>
</tr>
<tr>
<td>CIPVCLPV</td>
<td>Foreign Key validation for CI_COLL_PROC</td>
</tr>
<tr>
<td>CIPVCLSI</td>
<td>Insert CI_COLL_PROC_SA</td>
</tr>
<tr>
<td>CIPVCLSV</td>
<td>Foreign Key validation for CI_COLL_PROC_SA</td>
</tr>
<tr>
<td>CIPVCOI</td>
<td>Insert CI_COP_L</td>
</tr>
<tr>
<td>CIPVCOIL</td>
<td>Foreign Key validation for CI_COP_L</td>
</tr>
<tr>
<td>CIPVCOPI</td>
<td>Insert CI_COP</td>
</tr>
<tr>
<td>CIPVCOPK</td>
<td>Generate CI_COP keys</td>
</tr>
<tr>
<td>CIPVCRRI</td>
<td>Generate CI_CR_RAT_HIST keys</td>
</tr>
<tr>
<td>CIPVCRTI</td>
<td>Insert CI_CR_RAT_HIST</td>
</tr>
<tr>
<td>CIPVCRTV</td>
<td>Foreign Key validation for CI_CR_RAT_HIST</td>
</tr>
<tr>
<td>CIPVCSCI</td>
<td>Insert CI_CC</td>
</tr>
<tr>
<td>CIPVCSCV</td>
<td>Foreign Key validation for CI_CC</td>
</tr>
<tr>
<td>CIPVCVCI</td>
<td>Insert CI_COP_EVT_CHAR</td>
</tr>
<tr>
<td>CIPVCVCV</td>
<td>Foreign Key validation for CI_COP_EVT_CHAR</td>
</tr>
<tr>
<td>CIPVCVNI</td>
<td>Insert CI_COLL_EVT</td>
</tr>
<tr>
<td>CIPVCVNV</td>
<td>Foreign Key validation for CI_COLL_EVT</td>
</tr>
<tr>
<td>CIPVDCRI</td>
<td>Insert CI_DCL</td>
</tr>
<tr>
<td>CIPVDCCRK</td>
<td>Generate CI_DCL keys</td>
</tr>
<tr>
<td>CIPVDTCI</td>
<td>Insert CI_DV_TEST_CHAR</td>
</tr>
<tr>
<td>CIPVDTCV</td>
<td>Foreign Key validation for CI_DV_TEST_CHAR</td>
</tr>
<tr>
<td>CIPVDTMII</td>
<td>Insert CI_DV_TEST_COMP</td>
</tr>
<tr>
<td>CIPVDTMV</td>
<td>Foreign Key validation for CI_DV_TEST_COMP</td>
</tr>
<tr>
<td>CIPVDTRI</td>
<td>Insert CI_DV_TEST_RES</td>
</tr>
<tr>
<td>CIPVDTRV</td>
<td>Foreign Key validation for CI_DV_TEST_RES</td>
</tr>
<tr>
<td>CIPVDTTI</td>
<td>Insert CI_DV_TEST</td>
</tr>
<tr>
<td>CIPVDTTK</td>
<td>Generate CI_DV_TEST keys</td>
</tr>
<tr>
<td>CIPVDTTV</td>
<td>Foreign Key validation for CI_DV_TEST</td>
</tr>
<tr>
<td>CIPVFAI</td>
<td>Insert CI_FA</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CIPVFACK</td>
<td>Generate CI_FA keys</td>
</tr>
<tr>
<td>CIPVFAFV</td>
<td>Foreign Key validation for CI_FA</td>
</tr>
<tr>
<td>CIPVFAHI</td>
<td>Insert CI_FA_CHAR</td>
</tr>
<tr>
<td>CIPVFAHV</td>
<td>Foreign Key validation for CI_FA_CHAR</td>
</tr>
<tr>
<td>CIPVFALI</td>
<td>Insert CI_FA_LOG</td>
</tr>
<tr>
<td>CIPVFALV</td>
<td>Foreign Key validation for CI_FA_LOG</td>
</tr>
<tr>
<td>CIPVFARI</td>
<td>Insert CI_FA_REM</td>
</tr>
<tr>
<td>CIPVFARV</td>
<td>Foreign Key validation for CI_FA_REM</td>
</tr>
<tr>
<td>CIPVFORI</td>
<td>Insert CI_FO</td>
</tr>
<tr>
<td>CIPVFORK</td>
<td>Generate CI_FO keys</td>
</tr>
<tr>
<td>CIPVFORV</td>
<td>Foreign Key validation for CI_FO</td>
</tr>
<tr>
<td>CIPVFSTI</td>
<td>Insert CI_FA_STEP</td>
</tr>
<tr>
<td>CIPVFSTV</td>
<td>Foreign Key validation for CI_FA_STEP</td>
</tr>
<tr>
<td>CIPVFTFI</td>
<td>Insert CI_FT</td>
</tr>
<tr>
<td>CIPVFTTV</td>
<td>Foreign Key validation for CI_FT</td>
</tr>
<tr>
<td>CIPVFTGI</td>
<td>Insert CI_FT_GL</td>
</tr>
<tr>
<td>CIPVFTGV</td>
<td>Foreign Key validation for CI_FT_GL</td>
</tr>
<tr>
<td>CIPVFTPI</td>
<td>Insert CI_FT_PROC</td>
</tr>
<tr>
<td>CIPVFTPV</td>
<td>Foreign Key validation for CI_FT_PROC</td>
</tr>
<tr>
<td>CIPVFTXK</td>
<td>Generate CI_FT keys</td>
</tr>
<tr>
<td>CIPVIDSI</td>
<td>Insert CI_INTV_DATA_SET</td>
</tr>
<tr>
<td>CIPVIDSK</td>
<td>Generate CI_INTV_DATA_SET keys</td>
</tr>
<tr>
<td>CIPVIEQI</td>
<td>Insert CI_ITEM_EQ</td>
</tr>
<tr>
<td>CIPVIEQV</td>
<td>Foreign Key validation for CI_ITEM_EQ</td>
</tr>
<tr>
<td>CIPVILHI</td>
<td>Insert CI_ITEM_LOC_HIS</td>
</tr>
<tr>
<td>CIPVILHK</td>
<td>Generate CI_ITEM_LOC_HIS keys</td>
</tr>
<tr>
<td>CIPVILHV</td>
<td>Foreign Key validation for CI_ITEM_LOC_HIS</td>
</tr>
<tr>
<td>CIPVINLI</td>
<td>Insert CI_INTV_PF_L</td>
</tr>
<tr>
<td>CIPVINLV</td>
<td>Foreign Key validation for CI_INTV_PF_L</td>
</tr>
<tr>
<td>CIPVINPI</td>
<td>Insert CI_INTV_PF</td>
</tr>
<tr>
<td>CIPVINPK</td>
<td>Generate CI_INTV_PF keys</td>
</tr>
<tr>
<td>CIPVIRSI</td>
<td>Insert CI_REG_DATA_SET</td>
</tr>
<tr>
<td>CIPVIRSK</td>
<td>Generate CI_REG_DATA_SET keys</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>CIPVITCI</td>
<td>Insert CI_ITEM_CHAR</td>
</tr>
<tr>
<td>CIPVITCV</td>
<td>Foreign Key validation for CI_ITEM_CHAR</td>
</tr>
<tr>
<td>CIPVITTDI</td>
<td>Insert CI_INTV_DATA</td>
</tr>
<tr>
<td>CIPVITFFV</td>
<td>Foreign Key validation for CI_INTV_DATA</td>
</tr>
<tr>
<td>CIPVITEMI</td>
<td>Insert CI_ITEM</td>
</tr>
<tr>
<td>CIPVITTMK</td>
<td>Generate CI_ITEM keys</td>
</tr>
<tr>
<td>CIPVITTVI</td>
<td>Insert CI_INTV_VAL</td>
</tr>
<tr>
<td>CIPVITTVV</td>
<td>Foreign Key validation for CI_INTV_VAL</td>
</tr>
<tr>
<td>CIPVIVSI</td>
<td>Insert CI_INTV_VAL_SET</td>
</tr>
<tr>
<td>CIPVIVSK</td>
<td>Generate CI_INTV_VAL_SET keys</td>
</tr>
<tr>
<td>CIPVLLDI</td>
<td>Insert CI_LL_DETAIL</td>
</tr>
<tr>
<td>CIPVLLDV</td>
<td>Foreign Key validation for CI_LL_DETAIL</td>
</tr>
<tr>
<td>CIPVLNDI</td>
<td>Insert CI_LANDLORD</td>
</tr>
<tr>
<td>CIPVLNDK</td>
<td>Generate CI_LANDLORD keys</td>
</tr>
<tr>
<td>CIPVMEQI</td>
<td>Insert CI_MTR_EQ</td>
</tr>
<tr>
<td>CIPVMEQV</td>
<td>Foreign Key validation for CI_MTR_EQ</td>
</tr>
<tr>
<td>CIPVMIDI</td>
<td>Insert CI_MTR_ID</td>
</tr>
<tr>
<td>CIPVMIDV</td>
<td>Foreign Key validation for CI_MTR_ID</td>
</tr>
<tr>
<td>CIPVMLHI</td>
<td>Insert CI_MTR_LOC_HIS</td>
</tr>
<tr>
<td>CIPVMLHK</td>
<td>Generate CI_MTR_LOC_HIS keys</td>
</tr>
<tr>
<td>CIPVMLHV</td>
<td>Foreign Key validation for CI_MTR_LOC_HIS</td>
</tr>
<tr>
<td>CIPVMRCI</td>
<td>Insert CI_MR_CHAR</td>
</tr>
<tr>
<td>CIPVMRCV</td>
<td>Foreign Key validation for CI_MR_CHAR</td>
</tr>
<tr>
<td>CIPVMRDI</td>
<td>Insert CI_MR</td>
</tr>
<tr>
<td>CIPVMRDK</td>
<td>Generate CI_MR keys</td>
</tr>
<tr>
<td>CIPVMRDV</td>
<td>Foreign Key validation for CI_MR</td>
</tr>
<tr>
<td>CIPVMRMI</td>
<td>Insert CI_MR_REM</td>
</tr>
<tr>
<td>CIPVMRMV</td>
<td>Foreign Key validation for CI_MR_REM</td>
</tr>
<tr>
<td>CIPVMSGI</td>
<td>Insert CI_ACCT_MSG</td>
</tr>
<tr>
<td>CIPVMSGV</td>
<td>Foreign Key validation for CI_ACCT_MSG</td>
</tr>
<tr>
<td>CIPVMTCI</td>
<td>Insert CI_MTR_CHAR</td>
</tr>
<tr>
<td>CIPVMTCV</td>
<td>Foreign Key validation for CI_MTR_CHAR</td>
</tr>
<tr>
<td>CIPVMTGI</td>
<td>Insert CI_MTR_CONFIG</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>CIPVMTGK</td>
<td>Generate CI_MTR_CONFIG keys</td>
</tr>
<tr>
<td>CIPVMTRI</td>
<td>Insert CI_MTR</td>
</tr>
<tr>
<td>CIPVMTRK</td>
<td>Generate CI_MTR keys</td>
</tr>
<tr>
<td>CIPVNBSI</td>
<td>Insert CI_NB_SA</td>
</tr>
<tr>
<td>CIPVNBSV</td>
<td>Foreign Key validation for CI_NB_SA</td>
</tr>
<tr>
<td>CIPVNCDI</td>
<td>Insert CI_NCD</td>
</tr>
<tr>
<td>CIPVNCDV</td>
<td>Foreign Key validation for CI_NCD</td>
</tr>
<tr>
<td>CIPVNPMI</td>
<td>Insert CI_SA_NB_PARM</td>
</tr>
<tr>
<td>CIPVNPMV</td>
<td>Foreign Key validation for CI_SA_NB_PARM</td>
</tr>
<tr>
<td>CIPVNSPI</td>
<td>Insert CI_NB_SCHED_PAY</td>
</tr>
<tr>
<td>CIPVNSPK</td>
<td>Generate CI_NB_SCHED_PAY keys</td>
</tr>
<tr>
<td>CIPVNSPV</td>
<td>Foreign Key validation for CI_NB_SCHED_PAY</td>
</tr>
<tr>
<td>CIPVPAOI</td>
<td>Insert CI_PER_ADDR_OVRD</td>
</tr>
<tr>
<td>CIPVPAOV</td>
<td>Foreign Key validation for CI_PER_ADDR_OVRD</td>
</tr>
<tr>
<td>CIPVPAYI</td>
<td>Insert CI_PAY</td>
</tr>
<tr>
<td>CIPVPAYK</td>
<td>Generate CI_PAY keys</td>
</tr>
<tr>
<td>CIPVPAYV</td>
<td>Foreign Key validation for CI_PAY</td>
</tr>
<tr>
<td>CIPVPCHI</td>
<td>Insert CI_PREM_CHAR</td>
</tr>
<tr>
<td>CIPVPCHV</td>
<td>Foreign Key validation for CI_PREM_CHAR</td>
</tr>
<tr>
<td>CIPVPCNI</td>
<td>Insert CI_PER_CONTDET</td>
</tr>
<tr>
<td>CIPVPCNK</td>
<td>Generate CI_PER_CONTDET keys</td>
</tr>
<tr>
<td>CIPVPCNV</td>
<td>Foreign Key validation for CI_PER_CONTDET</td>
</tr>
<tr>
<td>CIPVPECI</td>
<td>Insert CI_PAY_EVT_CHAR</td>
</tr>
<tr>
<td>CIPVPECV</td>
<td>Foreign Key validation for CI_PAY_EVT_CHAR</td>
</tr>
<tr>
<td>CIPVPERI</td>
<td>Insert CI_PER</td>
</tr>
<tr>
<td>CIPVPERK</td>
<td>Generate CI_PERSON keys</td>
</tr>
<tr>
<td>CIPVPGOI</td>
<td>Insert CI_PREM_GEO</td>
</tr>
<tr>
<td>CIPVPGOV</td>
<td>Foreign Key validation for CI_PREM_GEO</td>
</tr>
<tr>
<td>CIPVPIDI</td>
<td>Insert CI_PER_ID</td>
</tr>
<tr>
<td>CIPVPIDV</td>
<td>Foreign Key validation for CI_PER_ID</td>
</tr>
<tr>
<td>CIPVPNMI</td>
<td>Insert CI_PER_NAME</td>
</tr>
<tr>
<td>CIPVPNMV</td>
<td>Foreign Key validation for CI_PER_NAME</td>
</tr>
<tr>
<td>CIPVPPEI</td>
<td>Insert CI_PER_PER</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>CIPVPPEV</td>
<td>Foreign Key validation for CI_PER_PER</td>
</tr>
<tr>
<td>CIPVPHI</td>
<td>Insert CI_PER_PHONE</td>
</tr>
<tr>
<td>CIPVPHHV</td>
<td>Foreign Key validation for CI_PER_PHONE</td>
</tr>
<tr>
<td>CIPVPRCI</td>
<td>Insert CI_PER_CHAR</td>
</tr>
<tr>
<td>CIPVRCV</td>
<td>Foreign Key validation for CI_PER_CHAR</td>
</tr>
<tr>
<td>CIPVPRMI</td>
<td>Insert CI_PREM</td>
</tr>
<tr>
<td>CIPVPRMK</td>
<td>Generate CI_PREM keys</td>
</tr>
<tr>
<td>CIPVPSAI</td>
<td>Insert CI_PER_ADDR_SEAS</td>
</tr>
<tr>
<td>CIPVPSAV</td>
<td>Foreign Key validation for CI_PER_ADDR_SEAS</td>
</tr>
<tr>
<td>CIPVPSGI</td>
<td>Insert CI_PAY_SEG</td>
</tr>
<tr>
<td>CIPVPSGK</td>
<td>Generate CI_PAY_SEG keys</td>
</tr>
<tr>
<td>CIPVPSGV</td>
<td>Foreign Key validation for CI_PAY_SEG</td>
</tr>
<tr>
<td>CIPVPYCI</td>
<td>Insert CI_PAY_CHAR</td>
</tr>
<tr>
<td>CIPVPYCV</td>
<td>Foreign Key validation for CI_PAY_CHAR</td>
</tr>
<tr>
<td>CIPVPYEI</td>
<td>Insert CI_PAY_EVENT</td>
</tr>
<tr>
<td>CIPVPYEK</td>
<td>Generate CI_PAY_EVENT keys</td>
</tr>
<tr>
<td>CIPVREDI</td>
<td>Insert CI_REG_DATA</td>
</tr>
<tr>
<td>CIPVREFV</td>
<td>Foreign Key validation for CI_REG_DATA</td>
</tr>
<tr>
<td>CIPVREGI</td>
<td>Insert CI_REG</td>
</tr>
<tr>
<td>CIPVREGK</td>
<td>Generate CI_REG keys</td>
</tr>
<tr>
<td>CIPVREGV</td>
<td>Foreign Key validation for CI_REG</td>
</tr>
<tr>
<td>CIPVRGCi</td>
<td>Insert CI_REG_CHAR</td>
</tr>
<tr>
<td>CIPVRGCV</td>
<td>Foreign Key validation for CI_REG_CHAR</td>
</tr>
<tr>
<td>CIPVREGRI</td>
<td>Insert CI_REG_READ</td>
</tr>
<tr>
<td>CIPVRRDV</td>
<td>Foreign Key validation for CI_REG_READ</td>
</tr>
<tr>
<td>CIPVRRDK</td>
<td>Generate CI_REG_READ keys</td>
</tr>
<tr>
<td>CIPVSACI</td>
<td>Insert CI_SA_CHAR</td>
</tr>
<tr>
<td>CIPVSACV</td>
<td>Foreign Key validation for CI_SA_CHAR</td>
</tr>
<tr>
<td>CIPVSAHI</td>
<td>Insert CI_SA_RS_HIST</td>
</tr>
<tr>
<td>CIPVSAHV</td>
<td>Foreign Key validation for CI_SA_RS_HIST</td>
</tr>
<tr>
<td>CIPVSAOI</td>
<td>Insert CI_SA_CONTERM</td>
</tr>
<tr>
<td>CIPVSAOV</td>
<td>Foreign Key validation for CI_SA_CONTERM</td>
</tr>
<tr>
<td>CIPVSAAPI</td>
<td>Insert CI_SA_SP</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CIPVSAPV</td>
<td>Foreign Key validation for CI_SA_SP</td>
</tr>
<tr>
<td>CIPVSAQI</td>
<td>Insert CI_SA_CONT_QTY</td>
</tr>
<tr>
<td>CIPVSAQV</td>
<td>Foreign Key validation for CI_SA_CONT_QTY</td>
</tr>
<tr>
<td>CIPVSARI</td>
<td>Insert CI_SA_RCHG_HIST</td>
</tr>
<tr>
<td>CIPVSARV</td>
<td>Foreign Key validation for CI_SA_RCHG_HIST</td>
</tr>
<tr>
<td>CIPVSACAI</td>
<td>Insert CI_SCM_ACCT</td>
</tr>
<tr>
<td>CIPVSCAV</td>
<td>Foreign Key validation for CI_SCM_ACCT</td>
</tr>
<tr>
<td>CIPVSCBV</td>
<td>Foreign Key validation for CI_SCM</td>
</tr>
<tr>
<td>CIPVSCCCI</td>
<td>Insert CI_SCM_CHAR</td>
</tr>
<tr>
<td>CIPVSCCV</td>
<td>Foreign Key validation for CI_SCM_CHAR</td>
</tr>
<tr>
<td>CIPVSCFI</td>
<td>Insert CI_SC_EVT_FT</td>
</tr>
<tr>
<td>CIPVSCFK</td>
<td>Generate CI_SC_EVT_FT keys</td>
</tr>
<tr>
<td>CIPVSCFV</td>
<td>Foreign Key validation for CI_SC_EVT_FT</td>
</tr>
<tr>
<td>CIPVSCMI</td>
<td>Insert CI_SCM</td>
</tr>
<tr>
<td>CIPVSCMK</td>
<td>Generate CI_SCM keys</td>
</tr>
<tr>
<td>CIPVSOCI</td>
<td>Insert CI_SA_COP_OVRD</td>
</tr>
<tr>
<td>CIPVSOCOV</td>
<td>Foreign Key validation for CI_SA_COP_OVRD</td>
</tr>
<tr>
<td>CIPVSOCI</td>
<td>Insert CI_SA_COP</td>
</tr>
<tr>
<td>CIPVSCPK</td>
<td>Generate CI_SA_COP keys</td>
</tr>
<tr>
<td>CIPVSCPV</td>
<td>Foreign Key validation for CI_SA_COP</td>
</tr>
<tr>
<td>CIPVSCVI</td>
<td>Insert CI_SC_EVT</td>
</tr>
<tr>
<td>CIPVSCVK</td>
<td>Generate CI_SC_EVT keys</td>
</tr>
<tr>
<td>CIPVSCCVV</td>
<td>Foreign Key validation for CI_SC_EVT</td>
</tr>
<tr>
<td>CIPVSECCI</td>
<td>Insert CI_SEV_EVT_CC</td>
</tr>
<tr>
<td>CIPVSECV</td>
<td>Foreign Key validation for CI_SEV_EVT_CC</td>
</tr>
<tr>
<td>CIPVSEDI</td>
<td>Insert CI_SEV_EVT_DEP</td>
</tr>
<tr>
<td>CIPVSEDV</td>
<td>Foreign Key validation for CI_SEV_EVT_DEP</td>
</tr>
<tr>
<td>CIPVSEFI</td>
<td>Insert CI_SEV_EVT_FA</td>
</tr>
<tr>
<td>CIPVSEFV</td>
<td>Foreign Key validation for CI_SEV_EVT_FA</td>
</tr>
<tr>
<td>CIPVSEG1</td>
<td>Insert CI_BSEG</td>
</tr>
<tr>
<td>CIPVSEGV</td>
<td>Foreign Key validation for CI_BSEG</td>
</tr>
<tr>
<td>CIPVSEPI</td>
<td>Insert CI_SEV_PROC</td>
</tr>
<tr>
<td>CIPVSEPK</td>
<td>Generate CI_SEV_PROC keys</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>CIPVSEPV</td>
<td>Foreign Key validation for CI_SEV_PROC</td>
</tr>
<tr>
<td>CIPVSEQI</td>
<td>Insert CI_SP_EQ</td>
</tr>
<tr>
<td>CIPVSEQV</td>
<td>Foreign Key validation for CI_SP_EQ</td>
</tr>
<tr>
<td>CIPVSEVI</td>
<td>Insert CI_SEV_EVT</td>
</tr>
<tr>
<td>CIPVSEVV</td>
<td>Foreign Key validation for CI_SEV_EVT</td>
</tr>
<tr>
<td>CIPVSIEI</td>
<td>Insert CI_SP_ITEM_EVT</td>
</tr>
<tr>
<td>CIPVSIEV</td>
<td>Foreign Key validation for CI_SP_ITEM_EVT</td>
</tr>
<tr>
<td>CIPVSIFV</td>
<td>Foreign Key validation for CI_SA_INTV_PF</td>
</tr>
<tr>
<td>CIPVSIIHI</td>
<td>Insert CI_SP_ITEM_HIST</td>
</tr>
<tr>
<td>CIPVSIIHK</td>
<td>Generate CI_SP_ITEM_HIST keys</td>
</tr>
<tr>
<td>CIPVSIIHV</td>
<td>Foreign Key validation for CI_SP_ITEM_HIST</td>
</tr>
<tr>
<td>CIPVSIPPI</td>
<td>Insert CI_SA_INTV_PF</td>
</tr>
<tr>
<td>CIPVSMEI</td>
<td>Insert CI_SP_MTR_EVT</td>
</tr>
<tr>
<td>CIPVSMEV</td>
<td>Foreign Key validation for CI_SP_MTR_EVT</td>
</tr>
<tr>
<td>CIPVSMGI</td>
<td>Insert CI_SA_MSG</td>
</tr>
<tr>
<td>CIPVSMGV</td>
<td>Foreign Key validation for CI_SA_MSG</td>
</tr>
<tr>
<td>CIPVSMHI</td>
<td>Insert CI_SP_MTR_HIST</td>
</tr>
<tr>
<td>CIPVSMHK</td>
<td>Generate CI_SP_MTR_HIST keys</td>
</tr>
<tr>
<td>CIPVSMHV</td>
<td>Foreign Key validation for CI_SP_MTR_HIST</td>
</tr>
<tr>
<td>CIPVSMIII</td>
<td>Insert CI_MULT_ITEM</td>
</tr>
<tr>
<td>CIPVSMIV</td>
<td>Foreign Key validation for CI_MULT_ITEM</td>
</tr>
<tr>
<td>CIPVSPCI</td>
<td>Insert CI_SP_CHAR</td>
</tr>
<tr>
<td>CIPVSPCV</td>
<td>Foreign Key validation for CI_SP_CHAR</td>
</tr>
<tr>
<td>CIPVSPIGI</td>
<td>Insert CI_SP_GEO</td>
</tr>
<tr>
<td>CIPVSPIGV</td>
<td>Foreign Key validation for CI_SP_GEO</td>
</tr>
<tr>
<td>CIPVSPMI</td>
<td>Insert CI_SP_MULT_ITEM</td>
</tr>
<tr>
<td>CIPVSPMV</td>
<td>Foreign Key validation for CI_SP_MULT_ITEM</td>
</tr>
<tr>
<td>CIPVSPOI</td>
<td>Insert CI_SP_OP_AREA</td>
</tr>
<tr>
<td>CIPVSPOV</td>
<td>Foreign Key validation for CI_SP_OP_AREA</td>
</tr>
<tr>
<td>CIPVSPPPI</td>
<td>Insert CI_SP</td>
</tr>
<tr>
<td>CIPVSPPPK</td>
<td>Generate CI_SP keys</td>
</tr>
<tr>
<td>CIPVSPPRI</td>
<td>Insert CI_SP_RTE</td>
</tr>
<tr>
<td>CIPVSPPRV</td>
<td>Foreign Key validation for CI_SP_RTE</td>
</tr>
<tr>
<td>Batch Control</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CIPVSQTI</td>
<td>Insert CI_BSEG_SQ</td>
</tr>
<tr>
<td>CIPVSQTV</td>
<td>Foreign Key validation for CI_BSEG_SQ</td>
</tr>
<tr>
<td>CIPVSRCl</td>
<td>Insert CI_SP_RTE_CFG</td>
</tr>
<tr>
<td>CIPVSRcv</td>
<td>Foreign Key validation for CI_SP_RTE_CFG</td>
</tr>
<tr>
<td>CIPVSRLI</td>
<td>Insert CI_SA_REL</td>
</tr>
<tr>
<td>CIPVSRLK</td>
<td>Generate CI_SA_REL keys</td>
</tr>
<tr>
<td>CIPVSRLV</td>
<td>Foreign Key validation for CI_SA_REL</td>
</tr>
<tr>
<td>CIPVSRRI</td>
<td>Insert CI_BSEG_READ</td>
</tr>
<tr>
<td>CIPVSRrv</td>
<td>Foreign Key validation for CI_BSEG_READ</td>
</tr>
<tr>
<td>CIPVScci</td>
<td>Insert CI_SA_SP_CHAR</td>
</tr>
<tr>
<td>CIPVSScv</td>
<td>Foreign Key validation for CI_SA_SP_CHAR</td>
</tr>
<tr>
<td>CIPVSSfi</td>
<td>Insert CI_SA_SP_FA</td>
</tr>
<tr>
<td>CIPVSSFv</td>
<td>Foreign Key validation for CI_SA_SP_FA</td>
</tr>
<tr>
<td>CIPVSSpk</td>
<td>Generate CI_SA_SP keys</td>
</tr>
<tr>
<td>CIPVStmi</td>
<td>Insert CI_SA_TOU_MAP</td>
</tr>
<tr>
<td>CIPVSTMv</td>
<td>Foreign Key validation for CI_SA_TOU_MAP</td>
</tr>
<tr>
<td>CIPVStai</td>
<td>Insert CI_SA</td>
</tr>
<tr>
<td>CIPVStak</td>
<td>Generate CI_SA keys</td>
</tr>
<tr>
<td>CIPVTbvi</td>
<td>Insert CI_TOU_BF_VAL</td>
</tr>
<tr>
<td>CIPVTbvv</td>
<td>Foreign Key validation for CI_TOU_BF_VAL</td>
</tr>
<tr>
<td>CIPVTCvi</td>
<td>Insert CI_TOU_CONT_VAL</td>
</tr>
<tr>
<td>CIPVTCvv</td>
<td>Foreign Key validation for CI_TOU_CONT_VAL</td>
</tr>
<tr>
<td>CIPVTdsi</td>
<td>Insert CI_TOU_DATA_SET</td>
</tr>
<tr>
<td>CIPVTdsk</td>
<td>Generate CI_TOU_DATA_SET keys</td>
</tr>
<tr>
<td>CIPVTmai</td>
<td>Insert CI_TOU_MAP</td>
</tr>
<tr>
<td>CIPVTmak</td>
<td>Generate CI_TOU_MAP keys</td>
</tr>
<tr>
<td>CIPVTmli</td>
<td>Insert CI_TOU_MAP_L</td>
</tr>
<tr>
<td>CIPVTmlv</td>
<td>Foreign Key validation for CI_TOU_MAP_L</td>
</tr>
<tr>
<td>CIPVtncl</td>
<td>Insert CI_PAY_TNDR_CHAR</td>
</tr>
<tr>
<td>CIPVtncv</td>
<td>Foreign Key validation for CI_PAY_TNDR_CHAR</td>
</tr>
<tr>
<td>CIPVTndi</td>
<td>Insert CI_PAY_TNDR</td>
</tr>
<tr>
<td>CIPVTndk</td>
<td>Generate CI_PAY_TNDR keys</td>
</tr>
<tr>
<td>CIPVTndv</td>
<td>Foreign Key validation for CI_PAY_TNDR</td>
</tr>
</tbody>
</table>
Validation Batch Control Application Services

The following batch controls use the C1-VALIDATE application service:

<table>
<thead>
<tr>
<th>Batch Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAL-ACCT</td>
<td>Validate account</td>
</tr>
</tbody>
</table>
### Purge Batch Control Application Services
The following batch controls use the C1-PURGE application service:

<table>
<thead>
<tr>
<th>Batch Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCUP-PRG</td>
<td>Purge billable charge upload</td>
</tr>
</tbody>
</table>
### ILM Batch Control Application Services

The following batch controls use the C1-ILM application service:

<table>
<thead>
<tr>
<th>Batch Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1-ADCRL</td>
<td>ILM Crawler - Adjustment</td>
</tr>
<tr>
<td>C1-BLCRL</td>
<td>ILM Crawler - Bill</td>
</tr>
<tr>
<td>C1-BSCRL</td>
<td>ILM Crawler - Bill Segment</td>
</tr>
<tr>
<td>C1-BCCRL</td>
<td>ILM Crawler - Billable Charge</td>
</tr>
<tr>
<td>C1-CACRL</td>
<td>ILM Crawler - Case</td>
</tr>
<tr>
<td>C1-CRCRL</td>
<td>ILM Crawler - Customer Relationship Request</td>
</tr>
<tr>
<td>C1-FACRL</td>
<td>ILM Crawler - Field Activity</td>
</tr>
<tr>
<td>C1-MECRL</td>
<td>ILM Crawler - Match Event</td>
</tr>
<tr>
<td>C1-MRCRL</td>
<td>ILM Crawler - Meter Read</td>
</tr>
<tr>
<td>C1-ORCRL</td>
<td>ILM Crawler - Order</td>
</tr>
<tr>
<td>C1-PECRL</td>
<td>ILM Crawler - Payment Event</td>
</tr>
<tr>
<td>C1-URCRL</td>
<td>ILM Crawler - Usage Request</td>
</tr>
</tbody>
</table>
This section describes database upgrades for the Oracle Utilities Meter Data Management v2.2.0.2 release, including:

- New Tables
- New Views
- Dropped Columns
- Added Columns
- Renamed Columns
- Column Format Change
- Primary Key Change
- Added Indexes
- Dropped Indexes
- Index Changes

### New Tables

<table>
<thead>
<tr>
<th>Table_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_IMD_CTRL_SEEDER</td>
</tr>
<tr>
<td>D1_STG_MSRMT</td>
</tr>
</tbody>
</table>

### New Views

There are no new tables in the Oracle Utilities Meter Data Management v2.2.0.2 release.

### Dropped Columns

No columns have been dropped in the Oracle Utilities Meter Data Management v2.2.0.2 release.

### Added Columns

<table>
<thead>
<tr>
<th>Table_Name</th>
<th>Column_Name</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_GTT3</td>
<td>D1_UOM_CD</td>
<td>N</td>
</tr>
<tr>
<td>D1_GTT3</td>
<td>INSTALL_OVRD_QTY</td>
<td>N</td>
</tr>
<tr>
<td>D1_US</td>
<td>MOST_RECENT_TRANS_DTTM</td>
<td>N</td>
</tr>
<tr>
<td>D1_USAGE</td>
<td>LINKED_TO_FRZN_BSEG_FLG</td>
<td>N</td>
</tr>
<tr>
<td>D1_USAGE</td>
<td>USED_ON_BILL_FLG</td>
<td>N</td>
</tr>
</tbody>
</table>
Renamed Columns

No columns have been renamed in the Oracle Utilities Meter Data Management v2.2.0.2 release.

Column Format Change

<table>
<thead>
<tr>
<th>Table/View Name</th>
<th>Column_Name</th>
<th>From</th>
<th>To</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_INBOUND_COMM_VW</td>
<td>CITY</td>
<td>VARCHAR2(30)</td>
<td>VARCHAR2(90)</td>
<td>VIEW</td>
</tr>
<tr>
<td>D1_OUTBOUND_COMM_VW</td>
<td>CITY</td>
<td>VARCHAR2(30)</td>
<td>VARCHAR2(90)</td>
<td>VIEW</td>
</tr>
<tr>
<td>D1_SP</td>
<td>CITY</td>
<td>VARCHAR2(30)</td>
<td>VARCHAR2(90)</td>
<td>TABLE</td>
</tr>
<tr>
<td>D1_SP</td>
<td>CITY_UPPER</td>
<td>VARCHAR2(30)</td>
<td>VARCHAR2(90)</td>
<td>TABLE</td>
</tr>
<tr>
<td>D1_USAGE_PERIOD_ITEM_DET</td>
<td>ITEM_SEQ_NUM</td>
<td>NUMBER(5)</td>
<td>NUMBER(3)</td>
<td>TABLE</td>
</tr>
</tbody>
</table>

Primary Key Change

There are no primary key changes in the Oracle Utilities Meter Data Management v2.2.0.2 release.

Added Indexes

<table>
<thead>
<tr>
<th>Table_Name</th>
<th>Index_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_IMD_CTRL_SEEDER</td>
<td>D1'T500P0</td>
</tr>
<tr>
<td>D1_IMD_CTRL_SEEDER</td>
<td>D1'T500S1</td>
</tr>
<tr>
<td>D1_STG_MSRMT</td>
<td>D1'T543P0</td>
</tr>
</tbody>
</table>
Dropped Indexes

<table>
<thead>
<tr>
<th>Table_Name</th>
<th>Index_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1_MSRMT</td>
<td>D1T298S1</td>
</tr>
<tr>
<td>D1_IMD_CTRL</td>
<td>D1T418S3</td>
</tr>
</tbody>
</table>

Index Changes

<table>
<thead>
<tr>
<th>From/To Table_Name</th>
<th>Index_Name</th>
<th>Column_Name</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>From D1_ACTIVITY</td>
<td>D1T319S0</td>
<td>BO_STATUS_CD</td>
<td>1</td>
</tr>
<tr>
<td>From D1_ACTIVITY</td>
<td>D1T319S0</td>
<td>BUS_OBJ_CD</td>
<td>2</td>
</tr>
<tr>
<td>From D1_ACTIVITY</td>
<td>D1T319S0</td>
<td>D1_ACTIVITY_ID</td>
<td>3</td>
</tr>
<tr>
<td>To D1_ACTIVITY</td>
<td>D1T319S0</td>
<td>BUS_OBJ_CD</td>
<td>1</td>
</tr>
<tr>
<td>To D1_ACTIVITY</td>
<td>D1T319S0</td>
<td>BO_STATUS_CD</td>
<td>2</td>
</tr>
<tr>
<td>To D1_ACTIVITY</td>
<td>D1T319S0</td>
<td>D1/activity_ID</td>
<td>3</td>
</tr>
<tr>
<td>From D1_USAGE</td>
<td>D1T281S1</td>
<td>BO_STATUS_CD</td>
<td>1</td>
</tr>
<tr>
<td>From D1_USAGE</td>
<td>D1T281S1</td>
<td>BUS_OBJ_CD</td>
<td>2</td>
</tr>
<tr>
<td>From D1_USAGE</td>
<td>D1T281S1</td>
<td>D1_USAGE_ID</td>
<td>3</td>
</tr>
<tr>
<td>To D1_USAGE</td>
<td>D1T281S1</td>
<td>BUS_OBJ_CD</td>
<td>1</td>
</tr>
<tr>
<td>To D1_USAGE</td>
<td>D1T281S1</td>
<td>BO_STATUS_CD</td>
<td>2</td>
</tr>
<tr>
<td>To D1_USAGE</td>
<td>D1T281S1</td>
<td>D1_USAGE_ID</td>
<td>3</td>
</tr>
</tbody>
</table>
Appendix L

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 v4.3.0.4 to v4.3.0.5
## Upgrading from Oracle Utilities Application Framework v4.3.0.4.0 to v4.3.0.5.0

### New Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Type of Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_DEPLOYMENT</td>
<td>Deployment</td>
<td>Transaction</td>
</tr>
<tr>
<td>F1_DEPLOYMENT_ITEM</td>
<td>Deployment Item</td>
<td>Transaction</td>
</tr>
<tr>
<td>F1_DEPLOYMENT_ITEM_METADATA</td>
<td>Deployment Item Meta Data</td>
<td>Transaction</td>
</tr>
<tr>
<td>F1_DEPLOYMENT_PART</td>
<td>Deployment Part</td>
<td>Master</td>
</tr>
<tr>
<td>F1_DEPLOYMENT_PART_L</td>
<td>Deployment Part Language</td>
<td>Master</td>
</tr>
<tr>
<td>F1_DEPLOYMENT_TYPE</td>
<td>Deployment Type</td>
<td>Master</td>
</tr>
<tr>
<td>F1_DEPLOYMENT_TYPE_L</td>
<td>Deployment Type Language</td>
<td>Master</td>
</tr>
<tr>
<td>F1_DEPTYP_DEPPART</td>
<td>Deployment Type / Deployment Part</td>
<td>Transaction</td>
</tr>
<tr>
<td>F1_DEPTYP_MDT_TYPE</td>
<td>Deployment Type / MDT Type</td>
<td>Transaction</td>
</tr>
<tr>
<td>F1_DEPTYP_MSG_CAT</td>
<td>Deployment Type Message Category</td>
<td>Transaction</td>
</tr>
<tr>
<td>F1_DEPTYP_USR_GRP</td>
<td>Deployment Type User Group</td>
<td>Transaction</td>
</tr>
<tr>
<td>F1_MDT</td>
<td>Mobile Data Terminal</td>
<td>Transaction</td>
</tr>
<tr>
<td>F1_MDT_CHAR</td>
<td>Mobile Data Terminal Characteristics</td>
<td>Transaction</td>
</tr>
<tr>
<td>F1_MDT_TYPE</td>
<td>Mobile Data Terminal Type</td>
<td>Master</td>
</tr>
<tr>
<td>F1_MDT_TYPE_CHAR</td>
<td>Mobile Data Terminal Type Characteristics</td>
<td>Master</td>
</tr>
<tr>
<td>F1_MDT_TYPE_L</td>
<td>Mobile Data Terminal Type Language</td>
<td>Master</td>
</tr>
<tr>
<td>F1_MOB_COMP_CHAR</td>
<td>Mobile Component Characteristics</td>
<td>Admin - System</td>
</tr>
<tr>
<td>F1_MOB_COMP_CNT</td>
<td>Mobile Component Content</td>
<td>Admin - System</td>
</tr>
<tr>
<td>F1_MOBILE_COMPONENT</td>
<td>Mobile Component</td>
<td>Admin - System</td>
</tr>
<tr>
<td>F1_MOBILE_COMPONENT_L</td>
<td>Mobile Component Language</td>
<td>Admin - System</td>
</tr>
</tbody>
</table>
Upgrades to the Oracle Utilities Application Framework Database

New Views
None

Dropped Tables

Table
F1_IWS_ANN_CHAR
F1_IWS_ANN_TYPE_CHAR

Unsupported Tables
None

Added Columns

Table | Column | Required
--- | --- | ---
CI_MD_SVC | APP_SVC_ID | N
F1_OUTMSG | BO_XMLDATA_AREA | N
F1_OUTMSG_TYPE | OUTMSG_PRIOR_FLG | Y
F1_OUTMSG_TYPE | OWNER_FLG | N
F1_OUTMSG_TYPE | TYPE_BUS_OBJ_CD | N

Note that in addition, the following table was added to 4.3.0.4.0 via a hot fix, but was not included in 4.3.0.5.0 until after the final build and is therefore added as a hot fix. Clients upgrading to 4.3.0.5.0 may see that the table is dropped via the blueprint and then reinstated after applying the bug fixes.
Dropped Columns
None

Unsupported Table Columns
None

Column Format Change
None

Primary Key Change
None

Index Changes
Index S1C675S1 for table F1_EXT_LOOKUP_VAL_CHAR has been renamed to F1C675S1.

<table>
<thead>
<tr>
<th>Table</th>
<th>Column</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1_OUTMSG_TYPE_L</td>
<td>OWNER_FLG</td>
<td>N</td>
</tr>
</tbody>
</table>
Appendix M

Oracle Utilities Customer To Meter
System Table Guide

This chapter lists the system tables owned by Oracle Utilities Customer To Meter V2.6.0.1.0 and explains the data standards of the system tables. The data standards are required for Oracle Utilities Customer To Meter installation, development within Oracle Utilities Customer To Meter, 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.

This section includes:

- Business Configuration Tables
Installation Options

The installation option has only one row that is shipped with the initial installation of Oracle Utilities Customer To Meter. The updateable 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.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>C0_INSTALLATION</td>
</tr>
</tbody>
</table>
| Initial Data | Create Field Activity Str Stop: Yes  
Premise Geo Type Usage: Required  
Alternate Representation: None  
CTI Integration : No  
Person ID Usage: Required  
Start Stop Detail Threshold : 30  
Bill Segment Freeze Option: Freeze At Will  
Accounting Date Freeze Option: Change If Period Is Closed  
Rollover Threshold Factor : 0.7  
User Can Override Bill Date : Yes  
Use High/Low Failures on Bill : Yes  
Base Time : 01/01/1900 02:00:00 AM  
Start Day Option: Current Day  
Use Alternative Bill ID : No (Yes, for upgrading customers that use Sequential Numbers)  
Alternative Bill ID Option : None ('Sequential Numbers', for upgrading customers that use Sequential Numbers)  
Use Credit Notes: No  
Autopay Creation Option: Create At Bill Completion  
Fund Accounting: Not Practiced  
Alternate Currency : Not Allowed |
This section lists the system tables owned by the Oracle Utilities Application Framework V4.3.0.5.0 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.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>SC_ACCESS_CNTL, SC_USER, SC_USR_GRP_PROF, SC_USR_GRP_USR, SC_USER_GROUP, SC_USER_GROUP_L</td>
</tr>
<tr>
<td>Initial Data</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_CURRENCY_CD, CI_CURRENCY_CD_L</td>
</tr>
<tr>
<td>Initial Data</td>
<td>United States Dollar (USD)</td>
</tr>
</tbody>
</table>

Display Profile

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

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_DISP_PROF, CI_DISP_PROF_L</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>F1_INSTALLATION, CI_INSTALL_ALG, CI_INSTALL_MSG, CI_INSTALL_MSG_L, CI_INSTALL_PROD</td>
</tr>
<tr>
<td>Initial Data</td>
<td>Option 11111</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_LANGUAGE</td>
</tr>
<tr>
<td>Initial Data</td>
<td>English (ENG)</td>
</tr>
</tbody>
</table>
Time Zone
The installation options require a valid time zone. A value for UTC (Coordinated Universal Time) is provided. Implementations should define the appropriate time zone and update the installation option value accordingly.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_TIME_ZONE, CI_TIME_ZONE_L</td>
</tr>
<tr>
<td>Initial Data</td>
<td>UTC</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_ROLE(L), CI_TD_VAL_ROLE</td>
</tr>
<tr>
<td>Initial Data</td>
<td>F1_DFLT</td>
</tr>
</tbody>
</table>

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 Oracle Utilities Customer Care and Billing 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, CI_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

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_ALG_TYPE, CI_ALG_TYPE_L, CI_ALG_TYPE_PRM, CI_ALG_TYPE_PRM_L</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Algorithm Type (ALG_TYPE_CD)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>None</td>
</tr>
</tbody>
</table>

## Algorithm

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_ALG, CI_ALG_L, CI_ALG_PARM, CI_ALG_VER</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Algorithm (ALG_CD)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>None</td>
</tr>
</tbody>
</table>

## Application Security

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>SC_APP_SERVICE, SC_APP_SERVICE_L, CI_APP_SVC_ACC</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Application Service ID (APP_SVC_ID).</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>None</td>
</tr>
</tbody>
</table>

## Batch Control

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_BATCH_CTRL, CI_BATCH_CTRL_L, CI_BATCH_CTRL_P, CI_BATCH_CTRL_P_L</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Batch Process (BATCH_CD), Program Name (PROGRAM_NAME)</td>
</tr>
</tbody>
</table>
### Properties | Description
--- | ---
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

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_CHAR_TYPE, CI_CHAR_TYPE_L, CI_CHAR_ENTITY, CI_CHAR_VAL, CI_CHAR_VAL_L.</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>F1_MIGR_PLAN, F1_MIGR_PLAN_L, F1_MIGR_PLAN_INSTR, F1_MIGR_PLAN_INSTR_L, F1_MIGR_PLAN_INSTR_ALG, F1_MIGR_REQ, F1_MIGR_REQ_L, F1_MIGR_REQ_INSTR, F1_MIGR_REQ_INSTR_L, F1_MIGR_REQ_INSTR_ENTITY, F1_MIGR_REQ_INCL_REQ</td>
</tr>
</tbody>
</table>

**Standard Data Fields**

- Migration Plan Code (MIGR_PLAN_CD), Migration Request Code (MIGR_REQ_CD)

**Customer Modification**

None

### Data Area

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>F1_DATA_AREA, F1_DATA_AREA_L</td>
</tr>
</tbody>
</table>

**Standard Data Fields**

- Data Area Code (DATA_AREA_CD)

**Customer Modification**

None

### Deployment Part

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>F1_DEPLOYMENT_PART, F1_DEPLOYMENT_PART_L, F1_DEPLOYMENT_ITEM</td>
</tr>
</tbody>
</table>

**Standard Data Fields**

- Deployment ID (F1_DEPLOYMENT_ID)

**Customer Modification**

None
### Display Icon

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_DISP_ICON, CI_DISP_ICON_L</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Display Icon Code (DISP_ICON_CD)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>None</td>
</tr>
</tbody>
</table>

### Extendable Lookup

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>F1_EXT_LOOKUP_VAL, F1_EXT_LOOKUP_VAL_L, F1_EXT_LOOKUP_VAL.Char</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Business Object (BUS_OBJ_CD), Extendable Lookup Value (F1_EXT_LOOKUP_VALUE)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>Business Object Data Area (BO_DATA_AREA) Override Description (DESCR_OVRD) on Extendable Lookup Field Value Language Table (F1_EXT_LOOKUP_VAL_L)</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_FK_REF, CI_FK_REF_L</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>FK reference code (FK_REF_CD)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>Info Program Name (INFO_PRG), Zone (ZONE_CD)</td>
</tr>
</tbody>
</table>

### Inbound Web Service

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>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</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Webservice Name (IN_SVC_NAME), Annotation (ANN_CD), Annotation Type (ANN_TYPE_CD)</td>
</tr>
</tbody>
</table>
### Legacy Object

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>F1_LGCY_OBJ</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Legacy Object ID (LGCY_OBJ_ID)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td></td>
</tr>
</tbody>
</table>

### Legacy Object Properties Description

- **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.

  For example: When the Oracle Utilities Customer Care and Billing product adds a new value to the algorithm entity flag (ALG_ENTITY_FLG), it is prefixed with C1.

### Lookup

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_LOOKUP_FIELD, CI_LOOKUP_VAL, CI_LOOKUP_VAL_L</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td></td>
</tr>
<tr>
<td>Customer Modification</td>
<td></td>
</tr>
</tbody>
</table>

### Lookup Properties Description

- **Override Description (DESCR_OVRD)** on Lookup Field Value Language Table (CI_LOOKUP_VAL_L)
### Map

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>F1_MAP, F1_MAP_L</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>UI Map (MAP_CD)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>None</td>
</tr>
</tbody>
</table>

### Managed Content

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>F1_MANAG_CONTENT, F1_MANAG_CONTENT_L</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Managed Content (MANAG_CONTENT_CD)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>None</td>
</tr>
</tbody>
</table>

### Messages

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_MSGCATEGORY, CI_MSGCATEGORY_L, CI_MSG, CI_MSG_L</td>
</tr>
</tbody>
</table>
### 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

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_MD_TBL, CI_MD_TBL_FLD, CI_MD_TBL_L, CI_MD_TBL_FLD_L, CI_MD_TBL_FLD_L, F1_DB_OBJECTS_REPO</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Table Name (TBL_NAME)</td>
</tr>
<tr>
<td></td>
<td>• Table names must match with the physical table name or view name in the database.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• 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)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>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.</td>
</tr>
</tbody>
</table>

Meta Data - Constraints

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_MD_CONST, CI_MD_CONST_FLD</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Constraint Id (CONST_ID)</td>
</tr>
<tr>
<td></td>
<td>• Index Name for Primary Constraints</td>
</tr>
<tr>
<td></td>
<td>• &lt;Index Name&gt;Rnn for Foreign Key Constraints Where</td>
</tr>
<tr>
<td></td>
<td>• nn: integer, 01 through 99</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>None</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>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</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Menu Name (MENU_NAME), Menu Item Id (MENU_ITEM_ID), Menu Line Id (MENU_LINE_ID)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>Override Label (OVRD_LABEL) on Menu Line Language Table (CI_MD_MENU_LINE_L)</td>
</tr>
</tbody>
</table>

**Meta Data - Program, Location and Services**

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>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</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Program Component Id (PROG_COM_ID), Location Id (LOC_ID), Program Component Name (PROG_COM_NAME), Service Name (SVC_NAME), Navigation Key (NAVIGATION_KEY)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>User Exit Program Name (USER_EXIT_PGM_NAME) on Program Components Table (CI_MD_PRG_COM),</td>
</tr>
</tbody>
</table>

**Meta Data - Maintenance Object**

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_MD_MO, CI_MD_MO_L, CI_MD_MO_TBL, CI_MD_MO_OPT, CI_MD_MO_ALG</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Maintenance Object (MAINT_OBJ_CD)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>None</td>
</tr>
</tbody>
</table>

**Meta Data - Work Tables**

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_MD_WRK_TBL, CI_MD_WRK_TBL_L, CI_MD_WRK_TBLFLD, CI_MD_MO_WRK</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Work Table Name (WRK_TBL_NAME)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>None</td>
</tr>
</tbody>
</table>
## Meta Data - Search Object

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>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</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Search Object (SO_CD)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>None</td>
</tr>
</tbody>
</table>

## Mobile Component

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>F1_MOBILE_COMPONENT, F1_MOBILE_COMPONENT_L, F1_MOB_COMP_CNT, F1_MOBILE_COMP_CHAR</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Mobile Component Code (F1_MOB_COMP_TYPE_CD)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>Expiration Days (F1_EXPIRATION_TIME_DUR)</td>
</tr>
</tbody>
</table>

## Navigation Option

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_NAV_OPT, CI_NAV_OPT_L, CI_NAV_OPT_CTXT, CI_NAV_OPT_USG, CI_MD_NAV</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Navigation Option Code (NAV_OPT_CD), Navigation Key (NAVIGATION_KEY)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>None</td>
</tr>
</tbody>
</table>

## Outbound Message Type

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>F1_OUTMSG_TYPE, F1_OUTMSG_TYPE_L</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Outbound Message Type Code (OUTMSG_TYPE_CD)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>Priority (OUTMSG_PRIOR_FLG)</td>
</tr>
</tbody>
</table>
## Portal and Zone

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>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</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Portal Code (PORTAL_CD), Zone Code (ZONE_CD), Zone Type Code (ZONE_HDL_CD)</td>
</tr>
<tr>
<td></td>
<td>• A new Zone can be added to the Product owned Portal Pages.</td>
</tr>
<tr>
<td></td>
<td>• The existing Zones cannot be removed from the Product owned Portal Pages.</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>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).</td>
</tr>
</tbody>
</table>

## Sequence

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_SEQ</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Sequence Name (SEQ_NAME)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>Sequence Number (SEQ_NBR)</td>
</tr>
<tr>
<td></td>
<td>This field is updated by the application process and must be set to 1 initially.</td>
</tr>
</tbody>
</table>

## Schema

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>F1_SCHEMA</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Schema Name (SCHEMA_NAME)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>None</td>
</tr>
</tbody>
</table>
### Script

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_SCR, CI_SCR_I, 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.</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Script (SCR_CD).</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>None</td>
</tr>
</tbody>
</table>

### To Do Type

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_TD_TYPE, CI_TD_TYPE_I, CI_TD_SRTKEY_TY, CI_TD_DRLKEY_TY, CI_TD_SRTKEY_TY_L.</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>To Do Type Code (TD_TYPE_CD).</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>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).</td>
</tr>
</tbody>
</table>

### Web Service Category

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>F1_WEB_CAT, F1_WEB_CAT_I, F1_WEB_CAT_INCL_SVC</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>Web Service Category code (WEB_SVC_CAT_CD).</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>None</td>
</tr>
</tbody>
</table>

### XAI Configuration

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_XAI_ADAPTER, CI_XAI_ADAPTER_I, CI_XAI_CLASS, CI_XAI_CLASS_I, CI_XAI_ENV_HNDL, CI_XAI_ENV_HNDL_I, CI_XAI_FORMAT, CI_XAI_FORMAT_I, CI_XAI_RCVR, CI_XAI_RCVR_I, CI_XAI_RCVR_CTX, CI_XAI_RCVR_RSP, CI_XAI_RCVR_RGRP, CI_XAI_SENDER, CI_XAI_SENDER_CTX, CI_XAI_SENDER_ID, CI_XAI_SNDR_CTX, CI_XAI_OPTION.</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>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).</td>
</tr>
</tbody>
</table>
### XAI Services

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Modification</td>
<td>Option Value (OPTION_VALUE) on Message Option Table (CI_XAI_OPTION)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>CI_XAI_IN_SVC, CI_XAI_IN_SVC_L, CI_XAI_SVC_PARM</td>
</tr>
<tr>
<td>Standard Data Fields</td>
<td>XAI Inbound Service Id (XAI_IN_SVC_ID), XAI Inbound Service Name (XAI_IN_SVC_NAME)</td>
</tr>
<tr>
<td>Customer Modification</td>
<td>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)</td>
</tr>
</tbody>
</table>
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. The product does not support customer specific data in these tables.

*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>
<thead>
<tr>
<th>Table Name</th>
<th>Upgrade Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_ALG</td>
<td>MG</td>
<td>Algorithm</td>
</tr>
<tr>
<td>CI_ALG_L</td>
<td>MG</td>
<td>Algorithm Language</td>
</tr>
<tr>
<td>CI_ALG_PARM</td>
<td>MG</td>
<td>Algorithm Parameters</td>
</tr>
<tr>
<td>CI_ALG_TYPE</td>
<td>MG</td>
<td>Algorithm Type</td>
</tr>
<tr>
<td>CI_ALG_TYPE_L</td>
<td>MG</td>
<td>Algorithm Type Language</td>
</tr>
<tr>
<td>CI_ALG_TYPE_PRM</td>
<td>MG</td>
<td>Algorithm Type Parameter</td>
</tr>
<tr>
<td>CI_ALG_TYPE_PRM_L</td>
<td>MG</td>
<td>Algorithm Type Parameter Language</td>
</tr>
<tr>
<td>CI_ALG_VER</td>
<td>MG</td>
<td>Algorithm Version</td>
</tr>
<tr>
<td>CI_APP_SVC_ACC</td>
<td>MG</td>
<td>Application Service Access Mode</td>
</tr>
<tr>
<td>CI_BATCH_CTRL</td>
<td>MG</td>
<td>Batch Control</td>
</tr>
<tr>
<td>CI_BATCH_CTRL_ALG</td>
<td>MG</td>
<td>Batch Control Algorithm</td>
</tr>
<tr>
<td>CI_BATCH_CTRL_L</td>
<td>MG</td>
<td>Batch Control Language</td>
</tr>
<tr>
<td>CI_BATCH_CTRL_P</td>
<td>MG</td>
<td>Batch Control Parameters</td>
</tr>
<tr>
<td>CI_BATCH_CTRL_P_L</td>
<td>MG</td>
<td>Batch Control Parameters Language</td>
</tr>
<tr>
<td>CI_CHAR_ENTITY</td>
<td>MG</td>
<td>Characteristic Type Entity</td>
</tr>
<tr>
<td>CI_CHAR_TYPE</td>
<td>MG</td>
<td>Characteristic Type</td>
</tr>
<tr>
<td>CI_CHAR_TYPE_L</td>
<td>MG</td>
<td>Characteristic Type Language</td>
</tr>
<tr>
<td>CI_CHAR_VAL</td>
<td>MG</td>
<td>Characteristic Type Value</td>
</tr>
<tr>
<td>CI_CHAR_VAL_L</td>
<td>MG</td>
<td>Characteristic Type Value Language</td>
</tr>
<tr>
<td>Table Name</td>
<td>Upgrade Action</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>CI_DISP_ICON</td>
<td>MG</td>
<td>Display Icon</td>
</tr>
<tr>
<td>CI_DISP_ICON_L</td>
<td>MG</td>
<td>Display Icon Language</td>
</tr>
<tr>
<td>CI_FK_REF</td>
<td>MG</td>
<td>Foreign Key Reference</td>
</tr>
<tr>
<td>CI_FK_REF_L</td>
<td>MG</td>
<td>Foreign Key Reference Language</td>
</tr>
<tr>
<td>CI_LANGUAGE</td>
<td>MG</td>
<td>Language Code</td>
</tr>
<tr>
<td>CI_LOOKUP_FIELD</td>
<td>MG</td>
<td>Lookup Field</td>
</tr>
<tr>
<td>CI_LOOKUP_VAL</td>
<td>MG</td>
<td>Lookup Field Value</td>
</tr>
<tr>
<td>CI_LOOKUP_VAL_L</td>
<td>MG</td>
<td>Lookup Field Value Language</td>
</tr>
<tr>
<td>CI_MD_CONST</td>
<td>MG</td>
<td>Constraints</td>
</tr>
<tr>
<td>CI_MD_CONST_FLD</td>
<td>MG</td>
<td>Constraint Fields</td>
</tr>
<tr>
<td>CI_MD_FLD</td>
<td>MG</td>
<td>Field</td>
</tr>
<tr>
<td>CI_MD_FLD_L</td>
<td>MG</td>
<td>Field Language</td>
</tr>
<tr>
<td>CI_MD_MENU</td>
<td>MG</td>
<td>Menu Information</td>
</tr>
<tr>
<td>CI_MD_MENU_IMOD</td>
<td>MG</td>
<td>Menu Item Module Maint</td>
</tr>
<tr>
<td>CI_MD_MENU_ITEM</td>
<td>MG</td>
<td>Menu Item</td>
</tr>
<tr>
<td>CI_MD_MENU_ITEM_L</td>
<td>MG</td>
<td>Menu Item Language</td>
</tr>
<tr>
<td>CI_MD_MENU_L</td>
<td>MG</td>
<td>Menu Language</td>
</tr>
<tr>
<td>CI_MD_MENU_LINE</td>
<td>MG</td>
<td>Menu Line</td>
</tr>
<tr>
<td>CI_MD_MENU_LINE_L</td>
<td>MG</td>
<td>Menu Line Language</td>
</tr>
<tr>
<td>CI_MD_MENU_MOD</td>
<td>MG</td>
<td>Menu Product Components</td>
</tr>
<tr>
<td>CI_MD_MO</td>
<td>MG</td>
<td>Maintenance Object</td>
</tr>
<tr>
<td>CI_MD_MO_ALG</td>
<td>MG</td>
<td>Maintenance Object Algorithm</td>
</tr>
<tr>
<td>CI_MD_MO_L</td>
<td>MG</td>
<td>Maintenance Object Language</td>
</tr>
<tr>
<td>CI_MD_MO_OPT</td>
<td>MG</td>
<td>Maintenance Object Option</td>
</tr>
<tr>
<td>CI_MD_MO_TBL</td>
<td>MG</td>
<td>Maintenance Object Table</td>
</tr>
<tr>
<td>CI_MD_MO_WRK</td>
<td>MG</td>
<td>Maintenance Object Work Tables</td>
</tr>
<tr>
<td>CI_MD_NAV</td>
<td>MG</td>
<td>Navigation Key</td>
</tr>
<tr>
<td>CI_MD_PRG_COM</td>
<td>MG</td>
<td>Program Components</td>
</tr>
<tr>
<td>CI_MD_PRG_ELEM</td>
<td>MG</td>
<td>UI Page Elements</td>
</tr>
<tr>
<td>CI_MD_PRG_EL_AT</td>
<td>MG</td>
<td>UI Page Element Attributes</td>
</tr>
<tr>
<td>CI_MD_PRG_LOC</td>
<td>MG</td>
<td>Program Location</td>
</tr>
<tr>
<td>CI_MD_PRG_MOD</td>
<td>MG</td>
<td>Program Module</td>
</tr>
<tr>
<td>Table Name</td>
<td>Upgrade Action</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>CI_MD_PRG_SEC</td>
<td>MG</td>
<td>UI Page Sections</td>
</tr>
<tr>
<td>CI_MD_PRG_SQL</td>
<td>MG</td>
<td>MD SQL Meta Data</td>
</tr>
<tr>
<td>CI_MD_PRG_TAB</td>
<td>MG</td>
<td>UI Tab Meta Data</td>
</tr>
<tr>
<td>CI_MD_PRG_VAR</td>
<td>MG</td>
<td>Program Variable</td>
</tr>
<tr>
<td>CI_MD_SO</td>
<td>MG</td>
<td>Search Object</td>
</tr>
<tr>
<td>CI_MD_SOCG</td>
<td>MG</td>
<td>Search Object Criteria Group</td>
</tr>
<tr>
<td>CI_MD_SOCG_FLD</td>
<td>MG</td>
<td>Search Object Criteria Group Field</td>
</tr>
<tr>
<td>CI_MD_SOCG_FLDAT</td>
<td>MG</td>
<td>Search Criteria Group Field Attribute</td>
</tr>
<tr>
<td>CI_MD_SOCG_L</td>
<td>MG</td>
<td>Search Object Criteria Group Language</td>
</tr>
<tr>
<td>CI_MD_SOCG_SORT</td>
<td>MG</td>
<td>Search Criteria Group Result Sort Order</td>
</tr>
<tr>
<td>CI_MD_SO_L</td>
<td>MG</td>
<td>Search Object Language</td>
</tr>
<tr>
<td>CI_MD_SO_RSFLD</td>
<td>MG</td>
<td>Search Object Result Field</td>
</tr>
<tr>
<td>CI_MD_SO_RSFLDAT</td>
<td>MG</td>
<td>Search Object Result Field Attribute</td>
</tr>
<tr>
<td>CI_MD_SVC</td>
<td>MG</td>
<td>MD Service</td>
</tr>
<tr>
<td>CI_MD_SVC_L</td>
<td>MG</td>
<td>MD Service Language</td>
</tr>
<tr>
<td>CI_MD_SVC_PRG</td>
<td>MG</td>
<td>MD Service Program</td>
</tr>
<tr>
<td>CI_MD_TAB_MOD</td>
<td>MG</td>
<td>UI Tab Module</td>
</tr>
<tr>
<td>CI_MD_TBL</td>
<td>MG</td>
<td>MD Table</td>
</tr>
<tr>
<td>CI_MD_TBL_FLD</td>
<td>MG</td>
<td>MD Table Field</td>
</tr>
<tr>
<td>CI_MD_TBL_FLD_L</td>
<td>MG</td>
<td>MD Table Field Language</td>
</tr>
<tr>
<td>CI_MD_TBL_L</td>
<td>MG</td>
<td>MD Table Language</td>
</tr>
<tr>
<td>CI_MD_WRK_TBL</td>
<td>MG</td>
<td>Work Table</td>
</tr>
<tr>
<td>CI_MD_WRK_TBLFLD</td>
<td>MG</td>
<td>Work Table Field</td>
</tr>
<tr>
<td>CI_MD_WRK_TBL_L</td>
<td>MG</td>
<td>Work Table Language</td>
</tr>
<tr>
<td>CI_MSG</td>
<td>MG</td>
<td>Message</td>
</tr>
<tr>
<td>CI_MSG_CATEGORY</td>
<td>MG</td>
<td>Message Category</td>
</tr>
<tr>
<td>CI_MSG_CATEGORY_L</td>
<td>MG</td>
<td>Message Category Language</td>
</tr>
<tr>
<td>CI_MSG_L</td>
<td>MG</td>
<td>Message Language</td>
</tr>
<tr>
<td>CI_NAV_OPT</td>
<td>MG</td>
<td>Navigation Option</td>
</tr>
<tr>
<td>CI_NAV_OPT_CTXT</td>
<td>MG</td>
<td>Navigation Option Context</td>
</tr>
<tr>
<td>CI_NAV_OPT_L</td>
<td>MG</td>
<td>Navigation Option Language</td>
</tr>
<tr>
<td>CI_NAV_OPT_USG</td>
<td>MG</td>
<td>Navigation Option Usage</td>
</tr>
<tr>
<td>Table Name</td>
<td>Upgrade Action</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>CI_PORTAL</td>
<td>MG</td>
<td>Portal</td>
</tr>
<tr>
<td>CI_PORTAL_L</td>
<td>MG</td>
<td>Portal Language</td>
</tr>
<tr>
<td>CI_PORTAL_OPT</td>
<td>MG</td>
<td>Portal Option</td>
</tr>
<tr>
<td>CI_PORTAL_ZONE</td>
<td>MG</td>
<td>Portal Zone</td>
</tr>
<tr>
<td>CI_SCR</td>
<td>MG</td>
<td>Script</td>
</tr>
<tr>
<td>CI_SCR_CRT</td>
<td>MG</td>
<td>Script Criteria</td>
</tr>
<tr>
<td>CI_SCR_CRT_GRP</td>
<td>MG</td>
<td>Script Criteria Group</td>
</tr>
<tr>
<td>CI_SCR_CRT_GRP_L</td>
<td>MG</td>
<td>Script Criteria Group Language</td>
</tr>
<tr>
<td>CI_SCR_DA</td>
<td>MG</td>
<td>Script Data Area</td>
</tr>
<tr>
<td>CI_SCR_FLD_MAP</td>
<td>MG</td>
<td>Script Field Mapping</td>
</tr>
<tr>
<td>CI_SCR_L</td>
<td>MG</td>
<td>Script Language</td>
</tr>
<tr>
<td>CI_SCR_PRMPT</td>
<td>MG</td>
<td>Script Prompt</td>
</tr>
<tr>
<td>CI_SCR_PRMPT_L</td>
<td>MG</td>
<td>Script Prompt Language</td>
</tr>
<tr>
<td>CI_SCR_STEP</td>
<td>MG</td>
<td>Script Step</td>
</tr>
<tr>
<td>CI_SCR_STEP_L</td>
<td>MG</td>
<td>Script Step Language</td>
</tr>
<tr>
<td>CI_SEQ</td>
<td>MG</td>
<td>Sequence</td>
</tr>
<tr>
<td>CI_TD_DRLKEY_TY</td>
<td>MG</td>
<td>To Do Type Drill Key</td>
</tr>
<tr>
<td>CI_TD_SRTKEY_TY</td>
<td>MG</td>
<td>To Do Type Sort Key</td>
</tr>
<tr>
<td>CI_TD_SRTKEY_TY_L</td>
<td>MG</td>
<td>To Do Type Sort Key Language</td>
</tr>
<tr>
<td>CI_TD_TYPE</td>
<td>MG</td>
<td>To Do Type</td>
</tr>
<tr>
<td>CI_TD_TYPE_L</td>
<td>MG</td>
<td>To Do Type Language</td>
</tr>
<tr>
<td>CI_UI_ZONE</td>
<td>MG</td>
<td>Context Sensitive Zone</td>
</tr>
<tr>
<td>CI_USR_NAV_LINK</td>
<td>MG</td>
<td>User Favorite Links</td>
</tr>
<tr>
<td>CI_XAI_ADAPTER</td>
<td>MG</td>
<td>XAI Adapter</td>
</tr>
<tr>
<td>CI_XAI_ADAPTER_L</td>
<td>MG</td>
<td>XAI Adapter Lang</td>
</tr>
<tr>
<td>CI_XAI_CLASS</td>
<td>MG</td>
<td>Message Class</td>
</tr>
<tr>
<td>CI_XAI_CLASS_L</td>
<td>MG</td>
<td>Message Class Language</td>
</tr>
<tr>
<td>CI_XAI_ENV_HNDL</td>
<td>MG</td>
<td>XAI Envelope Handler</td>
</tr>
<tr>
<td>CI_XAI_ENV_HNDL_L</td>
<td>MG</td>
<td>XAI Envelope Handler Language</td>
</tr>
<tr>
<td>CI_XAI_IN_SVC</td>
<td>MG</td>
<td>XAI Inbound Service</td>
</tr>
<tr>
<td>CI_XAI_IN_SVC_L</td>
<td>MG</td>
<td>XAI Inbound Service Language</td>
</tr>
<tr>
<td>CI_XAI_SVC_PARM</td>
<td>MG</td>
<td>XAI Inbound Service Parameters</td>
</tr>
<tr>
<td>Table Name</td>
<td>Upgrade Action</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>CI_ZONE</td>
<td>MG</td>
<td>Zone</td>
</tr>
<tr>
<td>CI_ZONE_HDL</td>
<td>MG</td>
<td>Zone Type</td>
</tr>
<tr>
<td>CI_ZONE_HDL_L</td>
<td>MG</td>
<td>Zone Type Language</td>
</tr>
<tr>
<td>CI_ZONE_HDL_PRM</td>
<td>MG</td>
<td>Zone Type Parameters</td>
</tr>
<tr>
<td>CI_ZONE_HDL_PRM_L</td>
<td>MG</td>
<td>Zone Type Parameters Language</td>
</tr>
<tr>
<td>CI_ZONE_L</td>
<td>MG</td>
<td>Zone Language</td>
</tr>
<tr>
<td>CI_ZONE_PRM</td>
<td>MG</td>
<td>Zone Parameters</td>
</tr>
<tr>
<td>F1_BUS_OBJ</td>
<td>MG</td>
<td>Business Object</td>
</tr>
<tr>
<td>F1_BUS_OBJ_ALG</td>
<td>MG</td>
<td>Business Object Algorithm</td>
</tr>
<tr>
<td>F1_BUS_OBJ_L</td>
<td>MG</td>
<td>Business Object Language</td>
</tr>
<tr>
<td>F1_BUS_OBJ_OPT</td>
<td>MG</td>
<td>Business Object Option</td>
</tr>
<tr>
<td>F1_BUS_OBJ_STATUS</td>
<td>MG</td>
<td>Business Object Status</td>
</tr>
<tr>
<td>F1_BUS_OBJ_STATUS_ALG</td>
<td>MG</td>
<td>Business Object Status Algorithm</td>
</tr>
<tr>
<td>F1_BUS_OBJ_STATUS_L</td>
<td>MG</td>
<td>Business Object Status Language</td>
</tr>
<tr>
<td>F1_BUS_OBJ_STATUS_OPT</td>
<td>MG</td>
<td>Business Object Status Option</td>
</tr>
<tr>
<td>F1_BUS_OBJ_STATUS_RSN</td>
<td>MG</td>
<td>Status Reason</td>
</tr>
<tr>
<td>F1_BUS_OBJ_STATUS_RSN_L</td>
<td>MG</td>
<td>Status Reason Language</td>
</tr>
<tr>
<td>F1_BUS_OBJ_TR_RULE</td>
<td>MG</td>
<td>Business Object Transition Rule</td>
</tr>
<tr>
<td>F1_BUS_OBJ_TR_RULE_L</td>
<td>MG</td>
<td>Business Object Transition Rule Language</td>
</tr>
<tr>
<td>F1_BUS_SVC</td>
<td>MG</td>
<td>Business Service</td>
</tr>
<tr>
<td>F1_BUS_SVC_L</td>
<td>MG</td>
<td>Business Service Language</td>
</tr>
<tr>
<td>F1_DATA_AREA</td>
<td>MG</td>
<td>Data Area</td>
</tr>
<tr>
<td>F1_DATA_AREA_L</td>
<td>MG</td>
<td>Data Area Language</td>
</tr>
<tr>
<td>F1_DB_OBJECTS_REPO</td>
<td>MG</td>
<td>Database Objects Repository</td>
</tr>
<tr>
<td>F1_DEPLOYMENT_ITEM</td>
<td>MG</td>
<td>Deployment Part Item</td>
</tr>
<tr>
<td>F1_DEPLOYMENT_PART</td>
<td>MG</td>
<td>Deployment Part</td>
</tr>
<tr>
<td>F1_DEPLOYMENT_PART_L</td>
<td>MG</td>
<td>Deployment Part Language</td>
</tr>
<tr>
<td>F1_EXT_LOOKUP_VAL</td>
<td>MG</td>
<td>Extendable Lookup</td>
</tr>
<tr>
<td>F1_EXT_LOOKUP_VAL_L</td>
<td>MG</td>
<td>Extendable Lookup Language</td>
</tr>
<tr>
<td>F1_EXT_LOOKUP_VAL_CHAR</td>
<td>MG</td>
<td>Extendable Lookup Characteristics</td>
</tr>
<tr>
<td>F1_IWS_ANN</td>
<td>MG</td>
<td>Web Service Annotation</td>
</tr>
<tr>
<td>Table Name</td>
<td>Upgrade Action</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>F1_IWS_ANN_L</td>
<td>MG</td>
<td>Web Service Annotation Language</td>
</tr>
<tr>
<td>F1_IWS_ANN_PARM</td>
<td>MG</td>
<td>Web Service Annotation Parameter</td>
</tr>
<tr>
<td>F1_IWS_ANN_TYPE</td>
<td>MG</td>
<td>Web Service Annotation Type</td>
</tr>
<tr>
<td>F1_IWS_ANN_TYPE_L</td>
<td>MG</td>
<td>Web Service Annotation Type Language</td>
</tr>
<tr>
<td>F1_IWS_ANN_TYPE_PARM</td>
<td>MG</td>
<td>Web Service Annotation Type Parm</td>
</tr>
<tr>
<td>F1_IWS_ANN_TYPE_PARM_L</td>
<td>MG</td>
<td>Web Service Annotation Type Parameter</td>
</tr>
<tr>
<td>F1_IWS_SVC</td>
<td>MG</td>
<td>Inbound Web Service</td>
</tr>
<tr>
<td>F1_IWS_SVC_L</td>
<td>MG</td>
<td>Inbound Web Service Language</td>
</tr>
<tr>
<td>F1_IWS_SVC_OPER</td>
<td>MG</td>
<td>Inbound Web Service Operations</td>
</tr>
<tr>
<td>F1_IWS_SVC_OPER_L</td>
<td>MG</td>
<td>Inbound Web Service Operations Language</td>
</tr>
<tr>
<td>F1_MANAG_CONTENT</td>
<td>MG</td>
<td>Managed Content</td>
</tr>
<tr>
<td>F1_MANAG_CONTENT_L</td>
<td>MG</td>
<td>Managed Content Language</td>
</tr>
<tr>
<td>F1_MAP</td>
<td>MG</td>
<td>UI Map</td>
</tr>
<tr>
<td>F1_MAP_L</td>
<td>MG</td>
<td>UI Map Language</td>
</tr>
<tr>
<td>F1_MIGR_PLAN</td>
<td>MG</td>
<td>Migration Plan</td>
</tr>
<tr>
<td>F1_MIGR_PLAN_INSTR</td>
<td>MG</td>
<td>Migration Plan Instruction</td>
</tr>
<tr>
<td>F1_MIGR_PLAN_INSTR_ALG</td>
<td>MG</td>
<td>Migration Plan Instruction Algorithm</td>
</tr>
<tr>
<td>F1_MIGR_PLAN_INSTR_L</td>
<td>MG</td>
<td>Migration Plan Instruction Language</td>
</tr>
<tr>
<td>F1_MIGR_PLAN_L</td>
<td>MG</td>
<td>Migration Plan Language</td>
</tr>
<tr>
<td>F1_MIGR_REQ</td>
<td>MG</td>
<td>Migration Request</td>
</tr>
<tr>
<td>F1_MIGR_REQ_INCL_REQ</td>
<td>MG</td>
<td>Migration Request Grouping</td>
</tr>
<tr>
<td>F1_MIGR_REQ_INSTR</td>
<td>MG</td>
<td>Migration Request Instruction</td>
</tr>
<tr>
<td>F1_MIGR_REQ_INSTR_ENTITTY</td>
<td>MG</td>
<td>Migration Request Instruction Entity</td>
</tr>
<tr>
<td>F1_MIGR_REQ_INSTR_L</td>
<td>MG</td>
<td>Migration Request Instruction Language</td>
</tr>
<tr>
<td>F1_MIGR_REQ_L</td>
<td>MG</td>
<td>Migration Request Language</td>
</tr>
<tr>
<td>F1_MOBILE_COMPONENT</td>
<td>MG</td>
<td>Mobile Component</td>
</tr>
<tr>
<td>F1_MOBILE_COMPONENT_L</td>
<td>MG</td>
<td>Mobile Component Language</td>
</tr>
<tr>
<td>F1_MOB_COMP_CHAR</td>
<td>MG</td>
<td>Mobile Component Characteristics</td>
</tr>
<tr>
<td>F1_MOB_COMP_CNT</td>
<td>MG</td>
<td>Mobile Component Content</td>
</tr>
<tr>
<td>Table Name</td>
<td>Upgrade Action</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>F1_OUTMSG_TYPE</td>
<td>MG</td>
<td>Outbound Message Type</td>
</tr>
<tr>
<td>F1_OUTMSG_TYPE_L</td>
<td>MG</td>
<td>Outbound Message Type Language</td>
</tr>
<tr>
<td>F1_SCHEMA</td>
<td>MG</td>
<td>Schema</td>
</tr>
<tr>
<td>F1_WEB_CAT</td>
<td>MG</td>
<td>Web Service Category</td>
</tr>
<tr>
<td>F1_WEB_CAT_L</td>
<td>MG</td>
<td>Web Service Category Language</td>
</tr>
<tr>
<td>F1_WEB_CAT_INCL_SVC</td>
<td>MG</td>
<td>Web Service Category Included Services</td>
</tr>
<tr>
<td>SC_ACCESS_CNTL</td>
<td>MG</td>
<td>User Group Access Control</td>
</tr>
<tr>
<td>SC_APP_SERVICE</td>
<td>MG</td>
<td>Application Service</td>
</tr>
<tr>
<td>SC_APP_SERVICE_L</td>
<td>MG</td>
<td>Application Service Language</td>
</tr>
<tr>
<td>SC_USR_GRP_PROF</td>
<td>MG</td>
<td>User Group Profile</td>
</tr>
<tr>
<td>CI_CURRENCY_CD</td>
<td>KP</td>
<td>Currency Code</td>
</tr>
<tr>
<td>CI_CURRENCY_CD_L</td>
<td>KP</td>
<td>Currency Code Language</td>
</tr>
<tr>
<td>CI_DISP_PROF</td>
<td>KP</td>
<td>Display Profile</td>
</tr>
<tr>
<td>CI_DISP_PROF_L</td>
<td>KP</td>
<td>Display Profile Language</td>
</tr>
<tr>
<td>CI_TIME_ZONE</td>
<td>KP</td>
<td>Time Zone</td>
</tr>
<tr>
<td>CI_TIME_ZONE_L</td>
<td>KP</td>
<td>Time Zone Language</td>
</tr>
<tr>
<td>CI_USR_PORTAL</td>
<td>KP</td>
<td>User Portal</td>
</tr>
<tr>
<td>CI_XAI_JNDI_SVR</td>
<td>KP</td>
<td>XAI JNDI Server</td>
</tr>
<tr>
<td>CI_XAI_JNDI_SVR_L</td>
<td>KP</td>
<td>XAI JNDI Server Language</td>
</tr>
<tr>
<td>CI_XAI_OPTION</td>
<td>KP</td>
<td>Message Option</td>
</tr>
<tr>
<td>CI_XAI_RCVR</td>
<td>KP</td>
<td>XAI Receiver</td>
</tr>
<tr>
<td>CI_XAI_RCVR_CTX</td>
<td>KP</td>
<td>XAI Receiver Context</td>
</tr>
<tr>
<td>CI_XAI_RCVR_L</td>
<td>KP</td>
<td>XAI Receiver Language</td>
</tr>
<tr>
<td>CI_XAI_RCVR_RGRP</td>
<td>KP</td>
<td>XAI Receiver Rule Group</td>
</tr>
<tr>
<td>CI_XAI_RCVR_RSP</td>
<td>KP</td>
<td>XAI Receiver Response</td>
</tr>
<tr>
<td>CI_XAI_SENDER</td>
<td>KP</td>
<td>Message Sender</td>
</tr>
<tr>
<td>CI_XAI_SENDER_L</td>
<td>KP</td>
<td>Message Sender Language</td>
</tr>
<tr>
<td>CI_XAI_SNDR_CTX</td>
<td>KP</td>
<td>Message Sender Context</td>
</tr>
<tr>
<td>F1_BUS_OBJ_STATUS_RSN_CHAR</td>
<td>KP</td>
<td>Status Reason Characteristic</td>
</tr>
<tr>
<td>F1_INSTALLATION</td>
<td>KP</td>
<td>Installation Option - Framework</td>
</tr>
<tr>
<td>SC_USER</td>
<td>KP</td>
<td>User</td>
</tr>
<tr>
<td>Table Name</td>
<td>Upgrade Action</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>SC_USER_CHAR</td>
<td>KP</td>
<td>User Characteristic</td>
</tr>
<tr>
<td>SC_USER_GROUP</td>
<td>KP</td>
<td>User Group</td>
</tr>
<tr>
<td>SC_USER_GROUP_L</td>
<td>KP</td>
<td>User Group Language</td>
</tr>
<tr>
<td>SC_USR_GRP_USR</td>
<td>KP</td>
<td>User Group User</td>
</tr>
<tr>
<td>CI_MD_ATT_TY</td>
<td>RF</td>
<td>MD Element Attribute Type</td>
</tr>
<tr>
<td>CI_MD_AT_DTL</td>
<td>RF</td>
<td>MD Element Attribute Type Detail</td>
</tr>
<tr>
<td>CI_MD_AT_DTL_L</td>
<td>RF</td>
<td>MD Element Attribute Type Detail  Language</td>
</tr>
<tr>
<td>CI_MD_CTL</td>
<td>RF</td>
<td>Generator Control</td>
</tr>
<tr>
<td>CI_MD_CTL_L</td>
<td>RF</td>
<td>Generator Control Language</td>
</tr>
<tr>
<td>CI_MD_CTL_TMPL</td>
<td>RF</td>
<td>Generator Control Template</td>
</tr>
<tr>
<td>CI_MD_ELTY</td>
<td>RF</td>
<td>MD Element Type</td>
</tr>
<tr>
<td>CI_MD_ELTY_AT</td>
<td>RF</td>
<td>Element Type Attributes</td>
</tr>
<tr>
<td>CI_MD_ELTY_L</td>
<td>RF</td>
<td>Element Type Language</td>
</tr>
<tr>
<td>CI_MD_LOOKUP_F</td>
<td>RF</td>
<td>MD Lookup Field</td>
</tr>
<tr>
<td>CI_MD_MSG</td>
<td>RF</td>
<td>MD Message</td>
</tr>
<tr>
<td>CI_MD_MSG_L</td>
<td>RF</td>
<td>MD Message Language</td>
</tr>
<tr>
<td>CI_MD_PDF</td>
<td>RF</td>
<td>Predefined Fields</td>
</tr>
<tr>
<td>CI_MD_PDF_VAL</td>
<td>RF</td>
<td>Predefined Values</td>
</tr>
<tr>
<td>CI_MD_SRC_TYPE</td>
<td>RF</td>
<td>Source Type</td>
</tr>
<tr>
<td>CI_MD_SRC_TYPE_L</td>
<td>RF</td>
<td>Source Type Language</td>
</tr>
<tr>
<td>CI_MD_TMPL</td>
<td>RF</td>
<td>Template</td>
</tr>
<tr>
<td>CI_MD_TMPL_ELTY</td>
<td>RF</td>
<td>Template Element Types</td>
</tr>
<tr>
<td>CI_MD_TMPL_L</td>
<td>RF</td>
<td>Template Language</td>
</tr>
<tr>
<td>CI_MD_TMPL_VAR</td>
<td>RF</td>
<td>Template Variable</td>
</tr>
<tr>
<td>CI_MD_TMPL_VAR_L</td>
<td>RF</td>
<td>Template Variable Language</td>
</tr>
<tr>
<td>CI_MD_VAR</td>
<td>RF</td>
<td>Variable</td>
</tr>
<tr>
<td>CI_MD_VAR_DTL</td>
<td>RF</td>
<td>Variable Detail</td>
</tr>
<tr>
<td>CI_MD_VAR_DTL_L</td>
<td>RF</td>
<td>Variable Detail Language</td>
</tr>
<tr>
<td>CI_XAI_EXECUTER</td>
<td>RF</td>
<td>XAI Executer</td>
</tr>
<tr>
<td>CI_XAI_EXECUTER_L</td>
<td>RF</td>
<td>XAI Executer Language</td>
</tr>
<tr>
<td>CI_XAI_FORMAT</td>
<td>RF</td>
<td>XAI Format</td>
</tr>
<tr>
<td>Table Name</td>
<td>Upgrade Action</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>CI_XAI_FORMAT_L</td>
<td>RF</td>
<td>XAI Format Language</td>
</tr>
<tr>
<td>F1_LGCY_OBJ</td>
<td>RF</td>
<td>Legacy Object</td>
</tr>
</tbody>
</table>