

# **Oracle Utilities Meter Data Management**

Installation and Configuration Guide

Release 1.6.1.23 for Windows

**E18211-24**

December 2018

(Revised April 2020)

Oracle Utilities Meter Data Management/Meter Data Management Installation and Configuration Guide,  
Volume 1, Release 1.6.1.23 for Windows

E18211-24

Copyright © 1999, 2018 Oracle and/or its affiliates. All rights reserved.

Primary Author: Lou Proseri

Contributor: Steve Pratt

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

## **NOTIFICATION OF THIRD-PARTY LICENSES**

Oracle Utilities software contains third party, open source components as identified below. Third-party license terms and other third-party required notices are provided below.

**License:** Apache 1.1

**Module:** xercesImpl.jar, xalan.jar

Copyright © 1999-2000 The Apache Software Foundation. All rights reserved.

Use of xercesImpl and xalan within the product is governed by the following (Apache 1.1):

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the disclaimer below. (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer below in the documentation and/or other materials provided with the distribution. (3) The end-user documentation included with the redistribution, if any, must include the following acknowledgment: “This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).” Alternately, this acknowledgment may appear in the software itself, if and

wherever such third-party acknowledgments normally appear. (4) Neither the component name nor Apache Software Foundation may be used to endorse or promote products derived from the software without specific prior written permission. (5) Products derived from the software may not be called “Apache”, nor may “Apache” appear in their name, without prior written permission.

THIS SOFTWARE IS PROVIDED “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**License:** Paul Johnston

**Modules:** md5.js

Copyright (C) Paul Johnston 1999 - 2002

Use of these modules within the product is governed by the following:

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the disclaimer below. (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer below in the documentation and/or other materials provided with the distribution. (3) Neither the component name nor the names of the copyright holders and contributors may be used to endorse or promote products derived from the software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**License:** Tom Wu

**Module:** jsbn library

Copyright © 2003-2005 Tom Wu. All rights reserved

Use of this module within the product is governed by the following:

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the disclaimer below. (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer below in the documentation and/or other materials provided with the distribution.

THE SOFTWARE IS PROVIDED “AS-IS” AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL TOM WU BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.



---

---

# Contents

## Contents

### What's New

<b>New Features in the Oracle Utilities Meter Data Management Installation and Configuration Guide .....</b>	<b>1-i</b>
New Features for Release 1.6.1.16.....	1-i
New Features for Release 1.6.1.0.....	1-ii
New Features for Release 1.6.0.0.....	1-ii

### Chapter 1

<b>Overview.....</b>	<b>1-1</b>
Installation and Configuration Overview .....	1-2
What is this book? .....	1-3

### Chapter 2

<b>Oracle Utilities Meter Data Management Network Environment and Database Installation .....</b>	<b>2-1</b>
The Oracle Utilities Meter Data Management Installation Package .....	2-2
Oracle Utilities Meter Data Management Network Environment .....	2-3
Client Workstation - C/S.....	2-4
Client Workstation - Web.....	2-5
Database Server.....	2-6
Application/Batch Processing Server.....	2-7
Web Server(s).....	2-8
Oracle Utilities Meter Data Management Database Installation.....	2-9
Installation Requirements .....	2-9
Installing the Database .....	2-9
Upgrading the Meter Data Management Database Schema .....	2-10
Updating the Reporting Framework for Oracle Business Intelligence Publisher.....	2-10
Verifying the Database.....	2-11
Recommendations for Improving Database Performance .....	2-13

### Chapter 3

<b>Installing the Oracle Utilities Meter Data Management Application Software.....</b>	<b>3-1</b>
Installing the Oracle Utilities Meter Data Management Software .....	3-2
Installing in Conjunction with the Energy Information Platform .....	3-2
Installing After Installation of the Energy Information Platform.....	3-2
Setting Up Configuration Files on the Web Server.....	3-3
MDM.CFG.XML Configuration File.....	3-5
Setting Up Configuration Files on Application Server and Workstations.....	3-8
Activating an MDMPORTAL Service .....	3-9
MDMPORTAL Services .....	3-9
Pre-Populated Records.....	3-9
Activating the LSMDM_WebPortal Service.....	3-10

### Chapter 4

<b>Setting Up Oracle Utilities Meter Data Management Database Tables .....</b>	<b>4-1</b>
Setting Up Lookup Tables and Data .....	4-2

Time-of-Use Schedules .....	4-2
Interval Extended Status Code .....	4-2
MDM Exception Type Table.....	4-4
MDM Exception Category Table.....	4-5
Read System Status Codes.....	4-5
Usage Category Table.....	4-5
Usage Exception Status Table.....	4-6
Usage Reading Status Table .....	4-6
Usage Type Table.....	4-6
Setting Up Oracle Utilities Meter Data Management Assets.....	4-8
Data Concentrator Table .....	4-8
Meter Manufacturers Table .....	4-8
Meter Read Systems Table.....	4-8
Meter Read System Exception Mapping Table .....	4-9
Meter Read System Interval Status Code Mapping Table .....	4-9
Meter Read System TOU Mapping Table.....	4-9
Meter Read System UOM Mapping Table.....	4-9
Physical Meter Table .....	4-10
Physical Meter Location History Table .....	4-10
Setting Up Meter Data Management Data.....	4-11
MDM Exception Codes Table.....	4-11
Meter Events Table .....	4-14
Meter Event Types Table .....	4-14
Physical Meter Statuses Table .....	4-15
Physical Meter Locations Table.....	4-15
Physical Meter Location Types Table.....	4-15
Physical Meter Types Table.....	4-15
Read Cycle Frequencies Table .....	4-16
Read Cycle Table.....	4-16
Read Cycle Dates Table .....	4-16
Setting Up Meter Data Management Validations and Meters.....	4-17
Validation Groups Table .....	4-17
MDM Meter Table.....	4-17
Meter Configuration Table .....	4-18
Setting Up Oracle Utilities Meter Data Management Bill Determinant Calculation Data .....	4-20
Account Table .....	4-21
Account History Table.....	4-22
BD Queue Type Table.....	4-22
BD Queue Table .....	4-22
BD Queue Archive Table.....	4-23
BD Request Table.....	4-23
Bill Determinants Table .....	4-24
Bill History Table .....	4-24
Bill History Values Table .....	4-25
Bill View Table .....	4-25
Billing Cycle Table .....	4-25
Billing Cycle Dates Table.....	4-25
Business Status Table .....	4-25
Customer Table.....	4-26
Meter Bill Determinant Values Table .....	4-26
Rate Code Table.....	4-27
Rate Code History Table .....	4-27
Rate Form Table .....	4-27
Rate Form Version Table .....	4-28
Service Point Table.....	4-28

Service Point Accounts Table .....	4-30
Service Point Bill Determinant Values Table .....	4-30
Meter Data Management Work Queues .....	4-31
Work Queue.....	4-31
Work Queue Type .....	4-31
Work Queue Hints.....	4-31
Work Queue Type Hint.....	4-32
Creating Custom Work Queue Records for use with Oracle Utilities Meter Data Management ...	4-33

## Chapter 5

<b>Setting Up Oracle Utilities Meter Data Management Processes .....</b>	<b>5-1</b>
Oracle Utilities Meter Data (.LSM) File Format.....	5-2
LSM Format Examples .....	5-2
LSM Format Element Descriptions.....	5-3
LSM Format Schema.....	5-7
Oracle Utilities Meter Data Management Business Rule Types.....	5-11
Usage Import Business Rules.....	5-11
Web Transaction Business Rules.....	5-16
Oracle Utilities Meter Data Management Data Converters.....	5-17
MV90 Interval Data Converter.....	5-17
MV9 Interval Data Converter.....	5-19
LSE Interval Data Converter .....	5-21
LegacyMV90 Interval Data Converter .....	5-22
Selecting the Appropriate Meter Data Management Interval Data Converter .....	5-23
Echelon Converter.....	5-25
CellNet Converter.....	5-27
MVRS Converter.....	5-29
Configuring Usage Data Import using the Adapter.....	5-33
Implementing Import/Usage Services and Business Rules .....	5-33
Import/Usage Business Rule Example .....	5-35
Implementing MDMElsterImport Services and Business Rules .....	5-36
MDMElsterImport Business Rule Example.....	5-37
Implementing MDMReady Services and Business Rules .....	5-38
MDMReady Business Rule Example - Usage Import.....	5-39
MDMReady Business Rule Example - Web Transactions .....	5-40
Configuring Partial Commit and Error Management of Import Files .....	5-41
Oracle Utilities Meter Data Management Adapter Properties.....	5-43
Oracle Utilities Meter Data Management Validations and Estimations .....	5-44
Critical Validations .....	5-44
Conditional Validations.....	5-45
Validations and Unit of Measure .....	5-46
Replacement Reading Validations .....	5-46
Out of Order Validations .....	5-48
Register Readings and Validations .....	5-49
Meter Changeout Validations.....	5-50
When and How Estimation is Performed.....	5-52
Interval Estimations .....	5-52
Consumption Estimations.....	5-56
Setting Up Automated Estimation Processing.....	5-57

## Chapter 6

<b>Setting Up Oracle Utilities Meter Data Management Billing Determinant Calculations.....</b>	<b>6-1</b>
Overview.....	6-2
The Bill Determinant Calculation Process.....	6-2
Understanding Bill Determinant Calculations .....	6-3
Database Tables .....	6-3

User Interface .....	6-5
Account Selector .....	6-6
Adapter Queue .....	6-11
Adapter Services and Business Rules.....	6-12
Calculator Engine.....	6-15
Configuring Bill Determinant Calculations .....	6-19
Setting Up Data.....	6-19
Configuring Adapter Services .....	6-20
Configuring Oracle Utilities Rules Language.....	6-23
Executing Bill Determinant Calculations using the AccountSelector Interface.....	6-40
Executing Bill Determinant Generation using the BDWS Web Service.....	6-46

## Chapter 7

<b>Configuring Oracle Utilities Meter Data Management Security .....</b>	<b>7-1</b>
Oracle Utilities Meter Data Management Security.....	7-2
Meter Data Management Features .....	7-2
Important Notes about Assigning Meter Data Management Permissions .....	7-3

## Appendix A

<b>Oracle Utilities Data Repository Meter Data Management Database Schema.....</b>	<b>A-1</b>
Oracle Utilities Meter Data Management Database Schema, page 1 .....	A-2
Oracle Utilities Meter Data Management Database Schema, page 2 .....	A-3

## Appendix B

<b>Integrating Oracle Utilities Meter Data Manager with a Customer Information System.....</b>	<b>B-1</b>
Overview.....	B-2
Synchronizing Account and Service Point Data .....	B-2
Processing Bill Determinant Calculation Requests.....	B-2
Processing Replacement Readings .....	B-2
Accessing Oracle Utilities Meter Data Management from External Applications .....	B-3
Synchronizing Account and Service Point Data.....	B-4
Synchronization Adapter Services and Business Rules .....	B-4
Synchronization Payload Formats.....	B-7
Configuring Synchronization Adapter Services .....	B-12
Synchronization Adapter RDL Schemas.....	B-14
Processing Bill Determinant Requests .....	B-20
Overview .....	B-20
Bill Determinant Request Adapter Services and Business Rules.....	B-21
Calculation Request XML Format .....	B-23
Saving Calculation Results to the Bill View Table .....	B-24
Bill Determinant Publisher .....	B-27
Configuring Bill Determinant Publisher Adapter Services.....	B-27
Processing Replacement Readings.....	B-32
Replacement Reading Adapter Services and Business Rules .....	B-32
Configuring Replacement Reading Adapter Services.....	B-34
Accessing Oracle Utilities Meter Data Management from External Applications.....	B-36
Configuring an Oracle Utilities Customer Care and Billing Dashboard Zone.....	B-36
Accessing Oracle Utilities Meter Data Management.....	B-37

## Appendix C

<b>Setting Up Oracle Utilities Meter Data Management Web Services .....</b>	<b>C-1</b>
Oracle Utilities Meter Data Management Web Services .....	C-2
Setting Up Oracle Utilities Meter Data Management Web Services.....	C-2
Oracle Utilities Meter Data Management Business Rules .....	C-4
Troubleshooting.....	C-8
Request and Response XML Schemas .....	C-8



# Index



# What's New

---

## New Features in the Oracle Utilities Meter Data Management Installation and Configuration Guide

This chapter outlines the new features of the 1.6.0 and 1.6.1 releases of Oracle Utilities Meter Data Management that are documented in this guide.

### New Features for Release 1.6.1.16

---

Feature	Description	For more information, refer to...
New MDMPORTAL service	A new MDMPORTAL runtime service type is used to execute upload, edit/save, validation, and approval operations performed via the web-based user interface. This service type replaces the Oracle Utilities Rules Language Server (LSRLS) which has been deprecated.	<b>Activating an MDMPORTAL Service</b> on page 3-9

---

## New Features for Release 1.6.1.0

Feature	Description	For more information, refer to...
Support for Integration with Oracle Utilities Customer Care and Billing	<p>This release includes support for integrating Oracle Utilities Meter Data Management and Oracle Utilities Customer Care and Billing to support the following business processes:</p> <ul style="list-style-type: none"> <li>• Synchronizing Account and Service Point Data</li> <li>• Processing Bill Determinant Calculation Requests</li> <li>• Processing Replacement Readings</li> <li>• Accessing Oracle Utilities Meter Data Management from Oracle Utilities Customer Care and Billing</li> </ul>	<b>Appendix B: Integrating Oracle Utilities Meter Data Manager with a Customer Information System</b>

## New Features for Release 1.6.0.0

Feature	Description	For more information, refer to...
Support for Billing Determinant Calculation	This release includes a framework for calculating billing determinants at the Account, Service Point, and Meter level.	<b>Chapter 6: Setting Up Oracle Utilities Meter Data Management Billing Determinant Calculations</b>
Import Usage RDL Performance Enhancements	Performance of the Import/Usage business rule type can be improved through use of the "calcreading" Business Rule property.	<b>Import/Usage Business Rule Properties</b> on page 5-12
Support for Meter Change Outs	<p>A "Meter Changeout" occurs when the associations between a physical and logical meter changes, and the current physical meter that is installed at a service point is swapped out for a new physical meter. This can occur when a physical meter is damaged or replaced.</p> <p>This release includes validations that can check for meter changeouts, check if readings overlap a meter changeout, and can calculate the start time and value for consumption and TOU readings that occur on or near meter changeout and that are missing the start time and/or start value.</p>	<p><b>Meter Changeout Validations</b> on page 5-50</p> <p>In the <i>Oracle Utilities Meter Data Management User's Guide</i>:</p> <p>All Rule Validations</p> <ul style="list-style-type: none"> <li>• <b>Meter Changeout Overlap Check</b> on page 3-19</li> </ul> <p>Consumption Rules</p> <ul style="list-style-type: none"> <li>• <b>Meter Changeout Start Time and Value</b> on page 3-26</li> </ul> <p>TOU Rules</p> <ul style="list-style-type: none"> <li>• <b>Meter Changeout Start Time and Value</b> on page 3-35</li> </ul>

Feature	Description	For more information, refer to...
New Interval Data Converters	<p>This release includes offers several new converters that can be used to convert interval data into the Oracle Utilities Meter Data (LSM) format.</p> <p>Usage import performance can be improved through use of these converters.</p>	<p><b>Oracle Utilities Meter Data Management Data Converters</b> on page 5-17, including:</p> <ul style="list-style-type: none"> <li>• <b>MV90 Interval Data Converter</b> on page 5-17</li> <li>• <b>MV9 Interval Data Converter</b> on page 5-19</li> <li>• <b>LSE Interval Data Converter</b> on page 5-21</li> <li>• <b>LegacyMV90 Interval Data Converter</b> on page 5-22</li> <li>• <b>Selecting the Appropriate Meter Data Management Interval Data Converter</b> on page 5-23</li> </ul>



# Chapter 1

---

## Overview

This chapter provides an overview of the installation and configuration of Oracle Utilities Meter Data Management, including:

- **Installation and Configuration Overview**
- **What is this book?**

## Installation and Configuration Overview

Installing and configuring Oracle Utilities Meter Data Management involves the following steps:

- Set up the Oracle Utilities Meter Data Management network environment as described in **Chapter 2: Oracle Utilities Meter Data Management Network Environment and Database Installation**.
- Install the Oracle Utilities Meter Data Management Database as described in **Chapter 2: Oracle Utilities Meter Data Management Network Environment and Database Installation** and **Chapter 3: Oracle Utilities Data Repository Schema Creation** of the *Oracle Utilities Energy Information Platform Installation Guide*.
- Install workstation and application server applications used by Oracle Utilities Meter Data Management on client machines as described in **Chapter 3: Installing the Oracle Utilities Meter Data Management Application Software**.
- Install Oracle Utilities Meter Data Management Web components on web server as described in **Chapter 3: Installing the Oracle Utilities Meter Data Management Application Software**.
- Set up and configure database records as described in **Chapter 4: Setting Up Oracle Utilities Meter Data Management Database Tables**.
- Set up and configure automated import and estimation processes as described in **Chapter 5: Setting Up Oracle Utilities Meter Data Management Processes**.
- Set up and configure automated and ad-hoc bill determinant calculation processes as described in **Chapter 6: Setting Up Oracle Utilities Meter Data Management Billing Determinant Calculations**.
- Set up and configure security for use with Oracle Utilities Meter Data Management as described in **Chapter 7: Configuring Oracle Utilities Meter Data Management Security**.
- Set up and configure Oracle Utilities Meter Data Management to work with a customer information system as described in **Appendix B: Integrating Oracle Utilities Meter Data Manager with a Customer Information System**.
- Set up and configure web services as described in **Appendix C: Setting Up Oracle Utilities Meter Data Management Web Services**.



---

## What is this book?

The *Oracle Utilities Meter Data Management Installation and Configuration Guide* describes how to install and configure Oracle Utilities Meter Data Management, including the following:

- **Chapter 1: Overview** (this chapter) provides an overview of the Oracle Utilities Meter Data Management installation and configuration process
- **Chapter 2: Oracle Utilities Meter Data Management Network Environment and Database Installation** describes how to install the Oracle Utilities Meter Data Management Database.
- **Chapter 3: Installing the Oracle Utilities Meter Data Management Application Software** describes how to install the Oracle Utilities Meter Data Management application software on an application server, on a web server, and on client machines.
- **Chapter 4: Setting Up Oracle Utilities Meter Data Management Database Tables** describes how to set up database tables used by Oracle Utilities Meter Data Management.
- **Chapter 5: Setting Up Oracle Utilities Meter Data Management Processes** describes how to set up automated import and estimation processes used by Oracle Utilities Meter Data Management.
- **Chapter 6: Setting Up Oracle Utilities Meter Data Management Billing Determinant Calculations** describes how to set up automated and ad-hoc bill determinant calculation processes used by Oracle Utilities Meter Data Management.
- **Chapter 7: Configuring Oracle Utilities Meter Data Management Security** describes how to configure security for use with Oracle Utilities Meter Data Management.
- **Appendix A: Oracle Utilities Data Repository Meter Data Management Database Schema** provides a database schema diagram of the database tables used with Oracle Utilities Meter Data Management.
- **Appendix B: Integrating Oracle Utilities Meter Data Manager with a Customer Information System** provides information about using Oracle Utilities Meter Data Management with a customer information system such as Oracle Utilities Customer Care and Billing.
- **Appendix C: Setting Up Oracle Utilities Meter Data Management Web Services** describes how to set up web services used by Oracle Utilities Meter Data Management.



# Chapter 2

---

## Oracle Utilities Meter Data Management Network Environment and Database Installation

This chapter describes the Oracle Utilities Meter Data Management network environment and how to install the Oracle Utilities Data Repository used by Oracle Utilities Meter Data Management, including:

- **The Oracle Utilities Meter Data Management Installation Package**
- **Oracle Utilities Meter Data Management Network Environment**
- **Oracle Utilities Meter Data Management Database Installation**

# The Oracle Utilities Meter Data Management Installation Package

There is a separate installation package for the Oracle Utilities Energy Information Platform and for each related product. The table below provides the installation package file names for the Oracle Utilities Energy Information Platform and Oracle Utilities Meter Data Management.

Product	Installation File Name
Oracle Utilities Energy Information Platform	1.6.1.xx.0.EIP.zip
Oracle Utilities Meter Data Management	1.6.1.xx.0.MDM.zip

The Oracle Utilities Meter Data Management installation package contains the following folders:

- **Install:** Contains the installation program for Oracle Utilities Meter Data Management, including:
  - Oracle Utilities MDM 1.6.1.xx.0.msi
- **DBScripts:** Contains the following database scripts for Oracle Utilities Meter Data Management:
  - Install: scripts used to create a new Meter Data Management schema
  - Upgrade: scripts used to update the Meter Data Management schema

See **Oracle Utilities Meter Data Management Database Installation** on page 2-9 for more information about installing and upgrading the Oracle Utilities Meter Data Management database schema.

- **Documentation:** Contains documentation for Oracle Utilities Meter Data Management, including:
  - Oracle Utilities Meter Data Management Installation and Configuration Guide
  - Oracle Utilities Meter Data Management User's Guide

**Note:** Oracle Utilities Meter Data Management documentation can also be downloaded separately.

---

# Oracle Utilities Meter Data Management Network Environment

This section contains options for hardware and software specifications for implementation of the Oracle Utilities Energy Information Platform and related products, including:

- Oracle Utilities Billing Component
- Oracle Utilities Load Analysis
- Oracle Utilities Load Profiling and Settlement
- Oracle Utilities Meter Data Management
- Oracle Utilities Quotations Management
- Oracle Utilities Rate Management
- Oracle Utilities Transaction Management

The Energy Information Platform application architecture comprises all of the components typically found in any n-tier enterprise architecture, including client machines, servers, and the supporting network infrastructure.

The Oracle Utilities Energy Information Platform uses five primary types of systems components. Suggested hardware specifications and required software for each component are listed below.

Before installing Oracle Utilities software you should consult with Oracle Utilities Consulting. They will make suggestions for your hardware and network architecture based on your business requirements. You should also consult with Oracle Utilities Consulting if your business requirements change, if your database grows significantly, or if you plan on migrating to a different version of Oracle Utilities software.

**Note:** Supported software versions are subject to change. Please refer to the Release Notes shipped with the software for the latest supported versions.

## Client Workstation - C/S

These are workstations used when running the client/server (C/S) versions of Oracle Utilities applications, such as Data Manager.

### Software for Client Workstation - C/S:

- Supported Operating Systems
  - Windows 8.1
  - Windows 10
- Supported Web Browsers
  - Microsoft Internet Explorer 11 with latest security updates
  - Microsoft Edge (Windows 10 only)
- Relational Database Management System (RDBMS) client with applicable ODBC driver and Data Provider for .NET

Note: When using 64-bit operating systems, the 32-bit version of the database client is required.

- Oracle Utilities Energy Information Platform software
- Microsoft .NET Framework 4.5
- Microsoft Visual C++ Redistributable Packages for Visual Studio 2013 (32-bit)
- Microsoft Data Access Component (MDAC) 2.8 or higher
- Microsoft XML Core Services (MSXML) 6.0 or higher

## Client Workstation - Web

These are workstations used when running the web-enabled versions of Oracle Utilities applications only.

### Software for Client Workstation - Web:

- Supported Operating Systems
  - Windows 8.1
  - Windows 10
- Supported Web Browsers
  - Microsoft Internet Explorer 11 with latest security updates
  - Microsoft Edge (Windows 10 only)

**Note:** For users that require access to both client/server and web browser-based applications, please refer to the requirements listed for "Client Workstations - Thick Client."

## Database Server

The database server houses the Oracle Utilities Data Repository. The hardware platform for the database server may be Windows/Intel or UNIX/RISC.

### Software for Database Server:

- Supported Operating Systems
  - Microsoft Windows Server 2012
  - Microsoft Windows Server 2016
  - LINUX (UNIX/Intel)
  - Sun Microsystems Solaris
  - HP-UX
  - IBM AIX (UNIX/RISC) software

Note: Refer to the *Oracle Utilities Energy Information Platform Quick Install Guide* for additional details about supported operating systems.

- RDBMS Server: Oracle

### Database Platforms:

- Oracle 11g, Release 2 (11.2.0.4)
- Oracle 12c, Release 1 (12.1.0.2 or 12.2.0.1)
- Oracle 19c, Release 1 (19.3.0.0.0)

**Note:** Supported database versions and drivers are subject to change. Please refer to the Oracle Release Notes shipped with the software for the latest compatibility matrix.



## Application/Batch Processing Server

Application servers are used when running Windows Services, such as those employed by the Reporting Framework and Adapter components. The application server(s) can also perform batch processes such as meter data upload and validation as well as batch billing and settlements.

### Software for the Application/Batch Server:

- Supported Operating Systems
  - Windows 2012 Server R2
  - Windows 2016 Server
- Supported Web Browsers
  - Microsoft Internet Explorer 11 with latest security updates
  - Microsoft Edge with latest security updates (Windows 2016 Server only)
- RDBMS client with applicable ODBC driver and Data Provider for .NET  
Note: When using 64-bit operating systems, the 32-bit version of the database client is required.
- Oracle Utilities Energy Information Platform software
- Microsoft .NET Framework 4.5
- Microsoft Visual C++ Redistributable Packages for Visual Studio 2013 (32-bit)
- Microsoft Data Access Component (MDAC) 2.8 or higher
- Microsoft XML Core Services (MSXML) 6.0 or higher
- Java Development Kit (JDK) 1.8 or later (for application servers that will be running the Adapter)

## Web Server(s)

Web servers run Microsoft Internet Information Service (IIS). These servers may be combined with the application server(s) for combination Web and Application servers.

### Software for Web Server(s):

- Supported Operating Systems
  - Windows 2012 R2 Enterprise Server with Microsoft Internet Information Service (IIS) 8.5
  - Windows 2016 Enterprise Server with Microsoft Internet Information Service (IIS) 10.0
- Supported Web Browsers
  - Microsoft Internet Explorer 11 with latest security updates
  - Microsoft Edge with latest security updates (Windows 2016 Server only)
- RDBMS client with applicable ODBC driver and Data Provider for .NET  
Note: When using 64-bit operating systems, the 32-bit version of the database client is required.
- Oracle Utilities Energy Information Platform software
- Microsoft .NET Framework 4.5
- Microsoft Visual C++ Redistributable Packages for Visual Studio 2013 (32-bit)
- Microsoft Data Access Component (MDAC) 2.8 or higher
- Microsoft XML Core Services (MSXML) 6.0 or higher

# Oracle Utilities Meter Data Management Database Installation

This section describes how to install and verify the Oracle Utilities Meter Data Management database tables and data in the Oracle Utilities Data Repository, including:

- **Installation Requirements**
- **Installing the Database**
- **Upgrading the Meter Data Management Database Schema**
- **Updating the Reporting Framework for Oracle Business Intelligence Publisher**
- **Verifying the Database**
- **Recommendations for Improving Database Performance**

**Note:** This section assumes that you have created the Oracle Utilities Data Repository schema as described in **Chapter 3: Oracle Utilities Data Repository Schema Creation** of the *Oracle Utilities Energy Information Platform Installation Guide*.

## Installation Requirements

The following are required in order to install the Oracle Utilities Meter Data Management tables and data into the Oracle Utilities Data Repository:

- The Oracle Utilities Data Repository (v1.6.1.0.0) schema must have been installed on the database instance on which you plan to run Oracle Utilities Meter Data Management.
- The **addMDM.cmd** file. This database script adds the Oracle Utilities Meter Data Management database tables and data to the Oracle Utilities Data Repository.
- The **updateMDM.cmd** file. This database script updates the tables and data used by Oracle Utilities Meter Data Management in the Oracle Utilities Data Repository. See **Upgrading the Meter Data Management Database Schema** on page 2-10 for more information.
- The **BIPMDM.cmd** file. This database script changes the pre-defined Meter Data Management reports from Crystal Reports to Oracle Business Intelligence Publisher. See **Updating the Reporting Framework for Oracle Business Intelligence Publisher** on page 2-10 for more information.

## Installing the Database

Installing the Oracle Utilities Meter Data Management database involves installing the Oracle Utilities Meter Data Management database schema into the Oracle Utilities Data Repository.

### Oracle Utilities Meter Data Management Schema

Open a command prompt and run the addMDM.CMD script. This script uses the following syntax:

```
addMDM [-d <database>] [-own <owner name>] -opw <owner password>
```

Parameter	Description
<database>	The name given to the instance as specified in the TNSNAMES.ORA file. This parameter is optional and if not specified, the script will connect to the default Oracle database.
<owner name>	The name of the user which will own the database objects. This parameter is optional. If not specified, the default user PWRLINE will own database objects.

Parameter	Description
<owner password>	The chosen password for the PWRLINE schema owner.

Like the base schema database tables and indexes, the Oracle Utilities Meter Data Management objects are created in the default tablespace of the PWRLINE user with default sizing parameters. If these defaults are required to be changed then the scripts may be edited.

## Upgrading the Meter Data Management Database Schema

If you are upgrading Oracle Utilities Meter Data Management from a previous version, you must upgrade Meter Data Management database schema. The following database upgrade scripts are included in the Meter Data Management installation package.

- The **updateMDM.cmd** file. This database script updates the meter data management tables and data used by Oracle Utilities Meter Data Management in the Oracle Utilities Data Repository. This script can only be used when upgrading the following schema version.

v1.6 Schema Version	v1.6.1 Schema Version
v1.6.0.0.0	v1.6.1.7.0

If you are upgrading from a different schema version, contact Oracle Global Customer Support.

## Upgrading Oracle Utilities Meter Data Management Schema

Open a command prompt and run the updateMDM.CMD script. This script uses the following syntax:

```
updateMDM [-d <database>] [-own <owner name>] -opw <owner password>
```

Parameter	Description
<database>	The name given to the instance as specified in the TNSNAMES.ORA file. This parameter is optional and if not specified, the script will connect to the default Oracle database.
<owner name>	The name of the user which will own Oracle Utilities database objects. This parameter is optional. If not specified, the default Oracle Utilities user PWRLINE will own database objects.
<owner password>	The chosen password for the PWRLINE schema owner.

## Updating the Reporting Framework for Oracle Business Intelligence Publisher

If you are using Oracle Business Intelligence Publisher for reporting, you must update the pre-defined reports in Reporting Framework database tables. The following database scripts are included in the Meter Data Management installation package for this purpose.

- The **BIPMDM.cmd** file. This script changes the pre-defined Meter Data Management reports from Crystal Reports to Oracle Business Intelligence Publisher.

## Oracle Utilities Meter Data Management Reports

Open a command prompt and run the BIPMDM.CMD script. This script uses the following syntax:

```
BIPMDM [-d <database>] [-own <owner name>] -opw <owner password>
```

Parameter	Description
<database>	The name given to the instance as specified in the TNSNAMES.ORA file. This parameter is optional and if not specified, the script will connect to the default Oracle database.
<owner name>	The name of the user which will own Oracle Utilities database objects. This parameter is optional. If not specified, the default Oracle Utilities user PWRLINE will own database objects.
<owner password>	The chosen password for the PWRLINE schema owner.

## Verifying the Database

Verifying the Oracle Utilities Meter Data Management database involves verifying the Oracle Utilities Meter Data Management database schema in the Oracle Utilities Date Repository.

### Verification - Meter Data Management Tables

To verify that the Oracle Utilities Meter Data Management schema tables are in place use the following procedure:

1. Log into the database using the PWRLINE user (Password =password).
2. Verify that the following tables exist in the database:
  - HUBINBOUNDTRANS
  - LSMDBDBUSINESSSTATUS
  - LSMDBDQUEUE
  - LSMDBDQUEUEARCHIVE
  - LSMDBDQUEUEEREQUEST
  - LSMDBDQUEUEUETYPE
  - LSMDBDVALUEMTR
  - LSMDEXCEPTION
  - LSMDEXCEPTIONCATEGORY
  - LSMDEXCEPTIONTYPE
  - LSMDEXCEPTSTATUS
  - LSMDLOCATION
  - LSMDLOCATIONTYPE
  - LSMDMTRDATACHANEXCEPT
  - LSMDMTRDATACHANHIST
  - LSMDMTRDATACHANNEL
  - LSMDMTRDATAACUT

- LSMDMTRDATAACUTV
- LSMDMTRDATAAREAD
- LSMDMTRDATAAREADTOU
- LSMDMTREVENT
- LSMDMTREVENTHIST
- LSMDMTREVENTTYPE
- LSMDMTRLOCATIONHIST
- LSMDMTRLOCHISTSTATUS
- LSMDMTRMANUFACTURER
- LSMDMTRREADSYSTEM
- LSMDMTRREADSYSHIST
- LSMDPHYSICALMETER
- LSMDPHYSMTRTYPE
- LSMDREADCYCLE
- LSMDREADCYCLEDATE
- LSMDREADCYCLEFREQ
- LSMDREADSTATUS
- LSMDUSAGECATEGORY
- LSMDUSAGETYPE
- LSSERVICEPOINT
- LSSRVCPOINTCLASS
- LSTOUPERIODNAME
- LSVVALIDATIONGROUP

**Appendix A: Oracle Utilities Data Repository Meter Data Management Database Schema** includes a diagram of the Oracle Utilities Meter Data Management database schema (v1.6.1.0.0) that provides details regarding the table and columns in the Oracle Utilities Meter Data Management schema, as well as the relationships between these tables in the Oracle Utilities Data Repository.

## Verification - Meter Data Management Reports

To verify that the meter data management reports have been updated, use the following procedure:

1. Log into the database using the PWRLINE user (Password =password).
2. Verify that the following records exist in the Report Instance table:

Report Name	Description	Report Type
MDM Adapter Interface Status	MDM Adapter Interface Status	BIPublisher
MDM Closed WQ Detail	MDM Closed WQ Detail	BIPublisher
MDM Physical Meter Events	MDM Physical Meter Events	BIPublisher
MDM Revenue Protection Usage Events	MDM Revenue Protection Usage Events	BIPublisher

Report Name	Description	Report Type
MDM Usage Exceptions	MDM Usage Exceptions	BIPublisher
MDM Missing Reads	MDM Missing Reads	BIPublisher
Meter Population Summary by Meter Read System	Meter Population Summary by Meter Read System	BIPublisher
Meter Population Summary by Read Cycle	Meter Population Summary by Read Cycle	BIPublisher
Process Control Summary	Process Control Summary by Date and Runtime Service	BIPublisher
Process Control Detail	Process Control Detail by Date	BIPublisher

## Recommendations for Improving Database Performance

Due the large volumes of data often processed in meter data management operations, database performance is an important issue to consider. This section addresses Oracle technologies that can be used to improve database performance. The following section describes the different methods of improving performance through database configurations.

### Oracle Secure Files

One of the most costly processes for Oracle Utilities Meter Data Management to perform is looking up historic data in a very large database. This is particularly costly when using validations that look up historical interval data such as the High Low Interval Validation. To improve performance related to looking up historic interval data, Oracle Utilities recommends using Oracle Secure Files with compression. When using this approach, Oracle Utilities recommends the following:

- Apply compression to the LOB column on the Meter Data Channel Cut (LSMDMTRDATAACUT) table
- Use a compression level of Medium.

In addition to the performance benefits with certain validations, Oracle Secure Files with compression can also significant reduce the overall size of the Meter Data Channel Cut (LSMDMTRDATAACUT) table.

### Oracle Database Partitions

Oracle Utilities recommends partitioning for any table greater than 100GB in size. Partitioning can enhance both database manageability and performance.

Oracle Utilities recommends partitioning large tables by date range (either quarterly or monthly), and then optionally sub partitioning by either hash value or list. Partitioning by date range will aid with archiving tasks as older partitions can be dropped when their data is no longer require to be available online. For queries that span multiple partitions, Oracle's parallel query option can be employed to improve performance. Database statistics can also be updated for individual partitions.

For Oracle Utilities Meter Data Management, Oracle Utilities specifically recommends applying partitioning to the Meter Data Read (LSMDMTRDATAREAD) and Meter Data Channel Cut (LSMDMTRDATAACUT) tables using a combination of date (monthly) range and hash value.

In addition, if Meter Data Management upload requests (input files from Adapter) per day are significantly large numbers (4 to 5 million) Oracle Utilities also recommends applying partitioning to the Inbound (HUBINBOUND) table. Archiving of any data table or partition will require the archival data from the Meter Data Read (LSMDMTRDATAREAD) and Meter Data Channel Cut (LSMDMTRDATAACUT) tables.

- For scalar usage stored in the Meter Data Read (LSMDMTRDATAREAD) table, the recommended partition is a monthly range partition on STOPREADTIME, and SUBPARTITION hash on UIDMTRDATACHANNEL
- For interval usage stored in the Meter Data Channel Cut (LSMDMTRDATACUT) table, the recommended partition is a monthly range partition on STARTTIME, and SUBPARTITION hash on UIDMTRDATACHANNEL
- For upload requests stored in the Inbound (HUBINBOUND) table, the recommended partition is a monthly range partition on RECEIVEDATE, and SUBPARTITION hash on UIDHUBINBOUND.
- For versioned interval usage stored in the Meter Data Channel Cut Version (LSMDMTRDATACUTV) table, the recommended partition is a monthly range partition on STARTTIME and SUBPARTITION hash on UIDMTRDATACHANNEL
- For exceptions stored in the MDM Exception (LSMDMTRDATAACHANEXCEPT) table, the recommended partition is a monthly range partition on STOPREADTIME, and SUBPARTITION hash on UIDMTRDATAACHANEXCEPT

Refer to the *Oracle Database 11g Release 2 (R2)* documentation for more information about the use of database compression and partitioning.



# Chapter 3

---

## Installing the Oracle Utilities Meter Data Management Application Software

This chapter describes how you install the Oracle Utilities Meter Data Management application software, including:

- **Installing the Oracle Utilities Meter Data Management Software**
- **Setting Up Configuration Files on the Web Server**
- **MDM.CFG.XML Configuration File**
- **Setting Up Configuration Files on Application Server and Workstations**
- **Activating an MDMPORTAL Service**

## Installing the Oracle Utilities Meter Data Management Software

This section describes how to install the Oracle Utilities Meter Data Management software including:

- **Installing in Conjunction with the Energy Information Platform**
- **Installing After Installation of the Energy Information Platform**

### Installing in Conjunction with the Energy Information Platform

To install the Oracle Utilities Meter Data Management software in conjunction with the Oracle Utilities Energy Information Platform, use the following procedure:

1. Navigate to the Install folder created by the Energy Information Platform installation package (1.6.1.xx.0.EIP.zip). This folder contains the following files:
  - Oracle Utilities EIP 1.6.1.xx.0.msi
  - setup.exe
2. Navigate to the Install folder created by the Oracle Utilities Meter Data Management installation package (1.6.1.xx.0.MDM.zip). This file contains the following files:
  - Oracle Utilities MDM 1.6.1.xx.0.msi
3. Move the "Oracle Utilities MDM 1.6.1.xx.0.msi" file into the same directory as the Energy Information Platform files.

The directory should now contain following:

- Oracle Utilities EIP 1.6.1.xx.0.msi
  - setup.exe
  - Oracle Utilities MDM 1.6.1.xx.0.msi
4. Double-click the setup.exe file.

A dialog opens asking you to confirm the products you wish to install. Click **Yes** to continue with the installation. Click **No** to cancel the installation.

5. Proceed with the installation (starting at Step 3) as outlined in **Chapter 4: Installing the Oracle Utilities Application Software** in the *Oracle Utilities Energy Information Platform Installation Guide*.

### Installing After Installation of the Energy Information Platform

To install the Oracle Utilities Meter Data Management software after the Oracle Utilities Energy Information Platform software has been installed, use the following procedure:

1. Navigate to the Install folder created by the Oracle Utilities Meter Data Management installation package (1.6.1.xx.0.MDM.zip). This file contains the following files:
  - Oracle Utilities MDM 1.6.1.xx.0.msi
2. Double-click the Oracle Utilities MDM 1.6.1.xx.0.msi file.

A dialog opens asking you to confirm the products you wish to install. Click **Yes** to continue with the installation. Click **No** to cancel the installation.

3. Proceed with the installation (starting at Step 3) as outlined in **Chapter 4: Installing the Oracle Utilities Application Software** in the *Oracle Utilities Energy Information Platform Installation Guide*.

---

# Setting Up Configuration Files on the Web Server

Oracle Utilities web applications use the following configuration files.

## LODESTAR.CFG

The LODESTAR.CFG file is a text file used to customize the working environment of the Oracle Utilities application software. See **LODESTAR.CFG** on page 2-2 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server.

**Note:** This file **MUST** be named LODESTAR.CFG on the web server.

## LSSECURE.CFG.XML

The LSSECURE.CFG.XML file specifies the data source used by the Security functionality and is required in order to run Oracle Utilities web-enabled applications. The LSSECURE.CFG.XML must be installed in the **C:\LODESTAR\CFG** directory. See **LSSECURE.CFG.XML** on page 2-42 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server.

**Note:** If an application server and web server are installed on different machines, the LSSECURE.CFG.XML file is installed in the **C:\LODESTAR\CFG** directory with the web server components.

## LSREPORTMONITOR.CFG.XML (optional)

The LSREPORTMONITOR.CFG.XML file specifies where report data is stored and how reports are processed through the web-enabled Oracle Utilities Energy Information Platform. The LSREPORTMONITOR.CFG.XML must be installed in the **C:\LODESTAR\CFG** directory. See **LSREPORTMONITOR.CFG.XML** on page 2-36 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server.

**Note:** If an application server and web server are installed on different machines, the LSREPORTMONITOR.CFG.XML file is installed in the **C:\LODESTAR\CFG** directory on both server.

## LSLOGGER.CFG.XML (optional)

The LSLOGGER.CFG.XML file defines how log files are generated by Oracle Utilities applications. Each Oracle Utilities component can be configured to write messages to a specified log file or an Event Log. The LSLOGGER.CFG.XML must be installed in the **C:\LODESTAR\CFG** directory. See **LSLOGGER.CFG.XML** on page 2-27 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server.

**Note:** If an application server and web server are installed on different machines, the LSLOGGER.CFG.XML file is installed in the **C:\LODESTAR\CFG** directory on both machines.

## LSSCHDLR.CFG.XML (optional)

The LSSCHDLR.CFG.XML file defines the data source(s) monitored by the Oracle Utilities Schedule Service (LSSCHDLR.exe). This service monitors the Scheduled Message (SCHEDULEDMESSAGE) table in each data source specified in this file. This file is required if the Oracle Utilities Schedule Service (LSSCHDLR.exe) is used. The LSSCHDLR.CFG.XML must

be installed in the **C:\LODESTAR\CFG** directory. See **LSSCHDLR.CFG.XML** on page 2-41 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server.

**Note:** If an application server and web server are installed on different machines, the LSSCHDLR.CFG.XML file is installed in the **C:\LODESTAR\CFG** directory on both machines.

### **LSRELAY.CFG.XML (optional)**

The LSRELAY.CFG.XML file identifies the machine running an SMTP service used to send email messages from the Oracle Utilities Energy Information Platform. This file is required only if the email functions are used. The LSRELAY.CFG.XML file must be installed in the **C:\LODESTAR\CFG** directory. See **LSRELAY.CFG.XML** on page 2-34 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server.

### **LOCALES.CFG.XML**

The LOCALES.CFG.XML file specifies the regional locales available to Oracle Utilities web-enabled applications. The LOCALES.CFG.XML file must be installed in the **C:\LODESTAR\Web\CCS** directory. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server. See **LOCALES.CFG.XML** on page 2-22 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about this file.

**Note:** If an application server and web server are installed on different machines, the LOCALES.CFG.XML file is installed in the **C:\LODESTAR\Web\ccs** directory with the web server components.

# MDM.CFG.XML Configuration File

In addition to the configuration files used by the Oracle Utilities Energy Information Platform, Oracle Utilities Meter Data Management also uses the MDM.CFG.XML file to define system level settings for Oracle Utilities Meter Data Management usage validations. The MDM.CFG.XML file must be installed in the **C:\LODESTAR\CFG** directory of any machine used to execute meter data management processing.

## MDM.CFG.XML Example

```
<MDM>
<IMPORT KEEPRAW="N" VALIDATE="Y" MDMERROR_DIR="c:\mdm\errors"
HOLIDAYLIST="MY_HOLIDAY_LIST" TOUSCHEDULE="MY_TOUSCHEDULE">
  <TIMEZONE SOURCE="CST" TARGET="CST"/>
  <DST PARTICIPANT="ASIS" MISSING="Y" INVALID="Reject"/>
  <WQ TYPE="USAGE" QUEUE="MDM" PRIORITYLEVEL="1" ASSIGNEDTOUSERID="MDM_USER"/>
</IMPORT>
<ESTIMATIONS DAYS="30" TOLERANCE="5" GAPSTARTDAYSBACK="30"
GAPSTOPDAYSBACK="1"/>
</MDM>
```

## MDM.CFG.XML Element Descriptions

Each of the data elements used by the MDM.CFG.XML file are described below.

**MDM:** Root element for the file.

**IMPORT:** Element that defines system level rules for import and validation of usage.

**Attributes:**

- **KEEPRAW:** Defines whether Oracle Utilities Meter Data Management should save a copy of each reading with a Usage Category of RAW. Valid values are “Y” (yes) and “N” (no). The default value is “N.”
- **VALIDATE:** Defines whether Oracle Utilities Meter Data Management should automatically validate readings upon import. Valid values are “Y” (yes) and “N” (no). The default value is “Y.”
- **MDMERRORDIR:** The path to a directory on the file system where error files are generated for critical exceptions.
- **HOLIDAYLIST:** The default holiday list defined in the Oracle Utilities Data Repository to be used by the Oracle Utilities Meter Data Management application.
- **TOUSCHEDULE:** The default Time of Use (TOU) Schedule defined in the Oracle Utilities Data Repository to be used by Oracle Utilities Meter Data Management.
- **REPLACEMENTOVERLAP:** defines how Oracle Utilities Meter Data Management will handle replacement readings. Valid values are:
  - **ALLOW:** Use the default rules for replacement readings (see **Replacement Reading Requirements** on page 5-46)
  - **REJECT:** Reject replacement readings unless they meet the following criteria:
    - The Start Time and Stop Time for the incoming reading exactly match a reading already in the database
    - The new reading matches exactly one and only one existing reading
    - If the new reading does not match exactly one cut, then it must be replacing the most recent reading for the meter. For example, if the last reading for a meter is from 2/1/06 through 2/28/06, if a new reading came in for 2/1/06 through 3/30/06, it would not be rejected.

- **REJECTMAN:** Use the default rules for replacement readings (see **Replacement Reading Requirements** on page 5-46) unless the reading being replaced has been manually edited, in which case the incoming reading will be rejected.
- **ESTCOMPVALUE:** The status code to which interval status codes are compared by the Oracle Utilities Meter Data Management application when identifying intervals to be estimated. For example, ESTCOMPVALUE="A" would compare interval status codes to "A". The default value for this attribute is "Z".
- **ESTCOMPOPERATION:** The type of comparison operation used by the Oracle Utilities Meter Data Management application when identifying intervals to estimate. For example, ESTCOMPOPERATION="GT" would estimate intervals whose status code is greater than the ESTCOMPVALUE attribute. Valid values include:
  - LT = Less Than (default)
  - LTE = Less Than or Equal to
  - GT = Greater Than
  - GTE = Greater Than or Equal to
  - EQUAL = Equal to
  - NOTEQUAL = Not Equal to
- **INTERVALRATE:** Optional name of the Oracle Utilities Rules Language rate schedule to execute for interval validations. The format of this attribute is:  
 <OPCO>:<JURIS>:<RATEFORMNAME>  
 where:
  - <OPCO> is the Operating Company for the rate form
  - <JURIS> is the Jurisdiction for the rate form
  - <RATEFORMNAME> is the Rate Form Name for the rate form
 The default value is: "MDM:MDM:MDMVALIDATIONS."
- **RATECONFIGFILENAME:** The path and file name for the LODESTAR.CFG configuration file used by the rate schedule specified in the INTERVALRATE element. Required only if the INTERVALRATE attribute is present.

#### Elements:

- **TIMEZONE:** Element that defines default time zone conversions for Oracle Utilities Meter Data Management. These default values can be overridden via the Timezone Conversion validation. See **Timezone Conversion** on page 3-21 in the *Oracle Utilities Meter Data Management User's Guide* for more information about this validation.

#### Attributes:

- **SOURCE:** The default source time zone.
- **TARGET:** The default target time zone.
- **DST:** Element that defines default settings for DST participant rules. These defaults can be overridden via the DST Participant Rules validation. See **DST Participant Rules** on page 3-21 in the *Oracle Utilities Meter Data Management User's Guide* for more information about this validation.

#### Attributes:

- **PARTICIPANT:** The default DST Participant flag. The default value is "ASIS."
- **MISSING:** The default value assigned to the DST\_Participant Flag when an incoming cut does not contain the flag. The default value is "Y."

- **INVALID:** The default value assigned to the DST\_Participant Flag when the incoming cut contain an invalid DST\_Participant flag. The default value is “Reject.”
- **WQ:** Element that defines default work queue properties for work queue items created by Oracle Utilities Meter Data Management.

**Attributes:**

- **TYPE:** The default Work Queue Type (WQTYPECODE) for work queue items created by Oracle Utilities Meter Data Management.
- **QUEUE:** The default Work Queue Type (WQQUEUECODE) for work queue items created by Oracle Utilities Meter Data Management.
- **PRIORITYLEVEL:** The default Work Queue Priority Level (PRIORITYLEVEL) for work queue items created by Oracle Utilities Meter Data Management.
- **ASSIGNEDTOUSERID:** The default User ID (ASSIGNEDTOUSERID) for work queue items created by Oracle Utilities Meter Data Management.

**ESTIMATIONS:** Element that defines rules for estimations performed by the MDMEstimations.exe program. This program scans for meters with missing reads based on Read Cycles, based on the values specified in this element.

**Attributes:**

- **DAYS:** The number of days in the past the system should check when identifying missing reads to estimate. For example, a setting of 6 would mean that the MDMEstimations.exe program would check the past 6 days for missing reads. The default value is 30.
- **TOLERANCE:** The number of days before a reading is considered “late.” For example, a setting of 5 would estimate readings for meters whose Read Cycles have Read Cycle Dates that are more than 5 days old, and would ignore meters whose Read Cycle Dates are within the last 5 days. The default value is 5.
- **GAPSTARTDAYSBACK:** The number of days before the current date to use as the start date when checking for gaps. Used to define a range of days used when checking for gaps.
- **GAPSTOPDAYSBACK:** The number of days before the current date to use as the stop date when checking for gaps. Used to define a range of days used when checking for gaps.

**READSYSTEMS:** Element that defines one or more meter read systems with which Oracle Utilities Meter Data Management will interact via web services.

**Elements:**

- **READSYSTEM:** Element that defines a single meter read system.

**Attributes:**

- **Name:** The name of the meter read system. This must match the Meter Read System column in the Meter Read System table.

**Elements:**

- **ACTION:** Element that defines a specific action to be invoked by a web service.

**Attributes:**

- **CODE:** The name of the action to be invoked. This must match the Action Code column in the Action table.
- **USERID:** The user ID used to access the external web service.
- **PASSWORD:** The password used to access the external web service. Note: this password **cannot** be encrypted via the EncryptCFG.exe program.
- **URL:** The URL of the external web service.

## Setting Up Configuration Files on Application Server and Workstations

You also need to set up the configuration files needed by application server and workstation applications such as Data Manager and batch executables.

### **LODESTAR.CFG**

The **LODESTAR.CFG** file is a text file used to customize the working environment of the Oracle Utilities application software. See **LODESTAR.CFG** on page 2-2 of the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file.

### **LSRELAY.CFG.XML (optional)**

The **LSRELAY.CFG.XML** file identifies the machine running an SMTP service used to send email messages from the Oracle Utilities Energy Information Platform. This file is required only if the email functions are used. The **LSRELAY.CFG.XML** file must be installed in the **C:\LODESTAR\CFG** directory on any application server or workstation used to process Rules Language configuration that uses the **EMAILCLIENT** function. See **LSRELAY.CFG.XML** on page 2-34 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server.



## Activating an MDMPORTAL Service

Oracle Utilities Meter Data Management uses the MDMPORTAL Adapter runtime service type to execute specific types of operations performed by users via the web-based user interface. This section describes how services of this type are used by Oracle Utilities Meter Data Management, and includes the following

- **MDMPORTAL Services**
- **Pre-Populated Records**
- **Activating the LSMDM\_WebPortal Service**

### MDMPORTAL Services

MDMPORTAL services (more specifically, runtime services of type "MDMPORTAL") execute operations performed by users working with the Oracle Utilities Meter Data Management application's web-based user interface. The specific types of operations executed by MDMPORTAL services include:

- Importing/uploading usage data (via LSM files) using the **Upload Files** screen
- Saving changes made to usage readings using the **Edit Usage, Edit TOU Usage, or Edit Values** screens.
- Manually validating and/or approving usage readings using the **Edit Usage, Edit TOU Usage, or Edit Values** screens.

In order to perform these operations using the web-based user interface, you **MUST** configure and activate at least one service of this type.

MDMPORTAL services require only two service properties by defined:

- **RULE\_NAME:** The name of the business rule the service will execute. MDMPORTAL services MDMReady business rules to execute the operations outlined above.
- **POLL\_INTERVAL:** The number of seconds between polling cycles (how frequently the service will poll for operations to execute). The default POLL\_INTERVAL for MDMPORTAL services is 2 seconds.

Multiple services of this type can be run at the same time and on the same application server. Any and all MDMPORTAL services poll operations to execute once activated and running. For performance reasons, if multiple users are using the web-based user interface to frequently perform the types of operations listed above (upload, saving, and manually validating/approving usage readings), Oracle Utilities recommends implementing multiple MDMPORTAL services.

### Pre-Populated Records

The Oracle Utilities Meter Data Management database includes pre-populated Runtime Service and Business Rule records that can be used. The details of these pre-populated records are as follows:

#### Runtime Service Record - LSMDM\_WebPortal

- **Name:** LSMDM\_WebPortal
- **JMS Queues:** N/A
- **JMS Filter:** N/A
- **JMS Parameters:** -server -Djava.endorsed.dirs=./lib
- **Origination System:** LTMH
- **Parameters:** -N/A

- **Internal Flag:** Y
- **Output Flag:** Y
- **Runtime Service Type:** MDMPORTAL
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME="RULE_NAME" VALUE="LSBR_MDMUSAGE" />
  <PROPERTY NAME="POLL_INTERVAL" VALUE="2" />
</PROPERTIES>
```

### **Business Rule Record - LSBR\_MDMUSAGE**

- **Class Name:** NULL
- **Rule Name:** LSBR\_MDMUSAGE
- **Description:** MDM Usage
- **Rule Type:** MDMReady
- **Time Expire:** 0

## **Activating the LSMDM\_WebPortal Service**

To use the pre-populated records described above to activate an MDMPORTAL service, create a Service Activation record as follows:

- **Server:** <SERVER\_NAME>
- **Runtime Services:** LSMDM\_WebPortal
- **Enabled:** Yes
- **Run at Startup:** Yes
- **Keep Alive:** Yes

# Chapter 4

---

## Setting Up Oracle Utilities Meter Data Management Database Tables

This chapter describes how to set up records in a number of tables in the Oracle Utilities Data Repository used by Oracle Utilities Meter Data Management, including:

- **Setting Up Lookup Tables and Data**
- **Setting Up Oracle Utilities Meter Data Management Assets**
- **Setting Up Meter Data Management Data**
- **Setting Up Meter Data Management Validations and Meters**
- **Setting Up Oracle Utilities Meter Data Management Bill Determinant Calculation Data**
- **Meter Data Management Work Queues**
- **Meter Data Management Work Queues**

You set up these records in the Oracle Utilities Data Repository using either the Data Manager or Data Navigator application, or the Oracle Utilities Meter Data Management user interface.

### **A Note about Predefined Records:**

Several of the tables described in this chapter contain predefined records used by the Oracle Utilities Meter Data Management application. These records are listed in the appropriate table description. **Do not modify or delete any of these predefined records.** Changes made to these records can result in errors when using the Oracle Utilities Meter Data Management application.

## Setting Up Lookup Tables and Data

Lookup tables and data are records used by other tables in the Oracle Utilities Data Repository required by Oracle Utilities Meter Data Management. Setting up lookup tables and data involves creating records in the following tables in the Oracle Utilities Data Repository:

- **Time-of-Use Schedules**
- **Interval Extended Status Code**
- **MDM Exception Type Table**
- **MDM Exception Category Table**
- **Read System Status Codes**
- **Usage Category Table**
- **Usage Exception Status Table**
- **Usage Reading Status Table**
- **Usage Type Table**

### Time-of-Use Schedules

Time-of-use (TOU) schedules are used to classify each hour in each day of the week (including holidays) in some way, such as “on peak,” “off peak,” or “shoulder,” or any other appropriate classification. TOU schedules allow different prices or charges to be applied to energy usage based on a specific day and time.

Time-of-use (TOU) periods are specific periods of time within a TOU schedule that define each hour in each day of the week (including holidays) by some classification such as “on peak,” “off peak,” or “shoulder,” or any appropriate classifications.

For example, time-of-use periods might define the hours of 9:00 PM (21:00:00) through 7:00 AM (07:00:00) on weekdays as “off peak” hours, and the hours of 7:00 AM through 9:00 PM on weekdays as “on peak.” In this case, energy used during the “on peak” hours could be charged a different (probably higher) rate than energy used during “off peak” hours.

Time-of-use schedules and periods are created using either the Data Manager application or the Oracle Utilities Energy Information Platform. See **Time-of-Use Schedules** on page 7-8 in the *Data Manager User's Guide* or **Time of Use Schedules** on page 7-14 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about creating Time-of-use schedules.

### Interval Extended Status Code

Records in this table represent extended status codes used with enhanced/generic interval data. Records in this table include the following information:

**UID:** Unique identifier for the extended status code.

**Code:** Abbreviation for the extended status code.

**Bit Position:** The bit position associated with the extended status code, used when encoding/decoding.

**Description:** The description of the extended status code.

**Type:** The type of extended status code, Interval or Channel.

This table is populated with the following pre-defined records:

<b>Code</b>	<b>Bit Position</b>	<b>Description</b>	<b>Type</b>
NA	0	Not Used	C

<b>Code</b>	<b>Bit Position</b>	<b>Description</b>	<b>Type</b>
AD	1	Added Interval (Data Correction)	C
RE	2	Replaced Interval (Data Correction)	C
ES	3	Estimated Interval (Data Correction)	C
OV	4	Pulse Overflow	C
HL	5	Data Out Of Limits	C
XC	6	Excluded Data	C
PY	7	Parity Error	C
TY	8	Energy Type (Register Changed)	C
LR	9	Alarm/Error	C
DI	10	Harmonic Distortion	C
IN	11	Point-to-Point Linear Interpolation	C
CK	12	Check Estimation	C
PO	0	Interval - Power Outage	I
SI	1	Interval - Short Interval (False For Mag Tape)	I
LI	2	Interval - Long Interval (Missing For Mag Tape)	I
CR	3	Interval - CRC Checksum Error	I
RA	4	Interval - RAM Checksum Error	I
RO	5	Interval - ROM Checksum Error	I
LA	6	Interval - Data Missing	I
CL	7	Interval - Hardware Clock Error	I
BR	8	Interval - Memory Reset	I
WT	9	Interval - Watchdog Timeout	I
TR	10	Interval - Time Reset	I
TM	11	Interval - Test Mode	I
LC	12	Interval - Load Control	I
NA	13	Interval - Not Used	I
AD	17	Channel - Added Interval (Data Correction)	I
RE	18	Channel - Replaced Interval (Data Correction)	I
ES	19	Channel - Estimated Interval (Data Correction)	I
OV	20	Channel - Pulse Overflow	I

Code	Bit Position	Description	Type
HL	21	Channel - Data Out Of Limits	I
XC	22	Channel - Excluded Data	I
PY	23	Channel - Parity Error	I
TY	24	Channel - Energy Type (Register Changed)	I
LR	25	Channel - Alarm/Error	I
DI	26	Channel - Harmonic Distortion	I
IN	27	Channel - Point-to-Point Linear Interpolation	I
CK	28	Channel - Check Estimation	I

Note: The above predefined records are used by Oracle Utilities Meter Data Management, and **should not be modified or deleted**. If the EXTENDEDSTATUSES.CFG.XML file is present in the C:\LODESTAR\CFG folder, it overrides settings in the Interval Extended Status Code table.

## MDM Exception Type Table

Records in the MDM Exception Type table represent specific types of MDM-related exceptions. Records in this table are used by the MDM Exception Codes table (see **MDM Exception Codes Table** on page 4-11). Records in this table include the following information:

- **Exception Type:** The name of the exception type
- **Description:** A description of the exception type

The MDM Exception Type table includes the following predefined records:

Exception Type	Description	Additional Information
ERROR	ERROR	Exceptions of type ERROR set the Reading Category of the reading to STAGING.
INFORMATIONAL	INFORMATIONAL	Exceptions of type INFORMATIONAL set the Reading Category of the reading to FINAL.

## MDM Exception Category Table

Records in the MDM Exception Category table represent specific categories of MDM-related exceptions. Records in this table are used by the MDM Exception Codes table (see **MDM Exception Codes Table** on page 4-11). Records in this table include the following information:

- **Exception Category:** The name of the exception category
- **Description:** A description of the exception category

The MDM Exception Category table includes the following predefined records:

Exception Category	Description	Additional Information
EXCEPTION	Exception	Can be viewed via Open Usage Exceptions
REVENUEPROTECTION	Revenue Protection Event	Can be viewed via Open Usage Exceptions or Revenue Protection Events

## Read System Status Codes

Records in the Read System Status Codes table define specific status codes that can apply to meter read systems. Records in this table include the following information:

- **Read System Status Code:** The status code
- **Description:** A description of the status code

## Usage Category Table

Records in the Usage Category table represent specific categories of usage. As usage records are imported into and validated by the Oracle Utilities Meter Data Management application, the Usage Category is updated to reflect the current status of the reading. Records in this table include the following information:

- **Category Code:** A code that designates the usage category
- **Description:** A description of the usage category

The Usage Category table includes the following predefined records:

Usage Category	Description
FINAL	Final Usage
INACTIVE	Inactive Usage
RAW	Raw Usage
STAGING	Staging Usage

## Usage Exception Status Table

Records in the Usage Exception Status table represent specific status of usage exceptions. Records in this table are used by the MDM Exceptions table. Records in this table include the following information:

- **Exception Status Code:** A code that designates the exception status code
- **Description:** A description of the exception status code

The Usage Exception Status table includes the following predefined records:

Exception Status Code	Description
APPROVED	Manually Approved
AUTO	Auto Approved
OPEN	Open
REPLACED	Usage Replaced
RESUBMITTED	Resubmitted

## Usage Reading Status Table

Records in the Usage Reading Status table represent specific status of usage. As usage records are imported into and validated by the Oracle Utilities Meter Data Management application, the Usage Reading Status is updated to reflect the current status of the reading. Records in this table include the following information:

- **Reading Status Code:** A code that designates the exception status code
- **Description:** A description of the exception status code

The Usage Reading Status table includes the following predefined records:

Reading Status Code	Description
FAILED	Failed
NEW	New
PROCESSED	Processed
VALID	Valid
VERIFIED	Verified

## Usage Type Table

Records in the Usage Type table represent specific types of usage. Records in this table include the following information:

- **Exception Status Code:** A code that designates the exception status code
- **Description:** A description of the exception status code

The Usage Type table includes the following predefined records:

Usage Type Code	Usage Type
CONSUMPTION	Consumption



---

<b>Usage Type Code</b>	<b>Usage Type</b>
INTERVAL	Interval
TOU	TOU

## Setting Up Oracle Utilities Meter Data Management Assets

Oracle Utilities Meter Data Management assets are records that define physical assets related to meters. Setting up Oracle Utilities Meter Data Management assets involves creating records in the following tables in the Oracle Utilities Data Repository:

- **Data Concentrator Table**
- **Meter Manufacturers Table**
- **Meter Read Systems Table**
- **Meter Read System Exception Mapping Table**
- **Meter Read System Interval Status Code Mapping Table**
- **Meter Read System TOU Mapping Table**
- **Meter Read System UOM Mapping Table**
- **Physical Meter Table**
- **Physical Meter Location History Table**

Refer to **Chapter 7: Assets and Setup** in the *Oracle Utilities Meter Data Management User's Guide* for more information about creating these records using the Oracle Utilities Meter Data Management user interface.

### Data Concentrator Table

Records in the Data Concentrator table represent devices used to collect information from Echelon meters. Records in this table include the following information:

- **UID:** A unique id for the data concentrator
- **Serial Number:** The serial number of the data concentrator
- **Model Number:** The manufacturer's model number for the data concentrator
- **Meter Manufacturer:** The manufacturer of the data concentrator
- **Notes:** Notes about the data concentrator

### Meter Manufacturers Table

Records in the Meter Manufacturer table represent manufacturers of meters used with the Oracle Utilities Meter Data Management application. Records in this table include the following information:

- **UID:** A unique id for the meter manufacturer
- **Meter Manufacturer:** The name of the meter manufacturer
- **Description:** A description of the meter manufacturer

### Meter Read Systems Table

Records in the Meter Read Systems table represent systems used for collection of meter reads. Records in this table include the following information:

- **UID:** A unique id for the meter read system
- **Meter Read System:** The name of the meter read system
- **Description:** A description of the meter read system

## Meter Read System Exception Mapping Table

Records in the Meter Read System Exception Mapping table map exceptions (from the MDM Exception Codes table) associated with a specific meter read systems (from the Meter Read Systems table) to alternate exception codes. Records in this table include the following information:

- **UID:** A unique id for the record
- **Meter Read System:** The name of the meter read system
- **Meter Data Exception:** The Meter Data Exception Code
- **Alternate Exception Code:** The alternate exception code

## Meter Read System Interval Status Code Mapping Table

Records in the Meter Read System Interval Status Code Mapping table map extended status codes (from the **Interval Extended Status Code** table) associated with a specific meter read systems (from the Meter Read Systems table) to alternate status codes. Records in this table include the following information:

- **UID:** A unique id for the meter read system
- **Meter Read System:** The name of the meter read system
- **Extended Status:** The extended status code
- **Alternate Status Code:** The alternate status code
- **Priority:** The priority of the mapping. In the case of multiple mappings to the same alternate code, this column is used to determine which status code to display. The lower the number, the higher the priority.

## Meter Read System TOU Mapping Table

Records in the Meter Read System TOU Mapping table map time of use periods (from the TOU Period table) associated with a specific meter read systems (from the Meter Read Systems table) to alternate time of use period names. Records in this table include the following information:

- **UID:** A unique id for the meter read system
- **Meter Read System:** The name of the meter read system
- **TOU Period:** The time of use period
- **Alternate Period Name:** The alternate TOU period name

## Meter Read System UOM Mapping Table

Records in the Meter Read System UOM Mapping table map unit of measure codes (from the Unit of Measure table) associated with a specific meter read systems (from the Meter Read Systems table) to alternate unit of measure names. Records in this table include the following information:

- **UID:** A unique id for the meter read system
- **Meter Read System:** The name of the meter read system
- **Unit of Measure:** The unit of measure (from the Unit of Measure table)
- **Alternate UOM Name:** The alternate UOM name

## Physical Meter Table

Records in the Physical Meter table represent actual physical meters that record usage and consumption. Records in this table include the following information:

- **UID:** A unique id for the physical meter
- **Serial Number:** The serial number of the meter
- **Meter Manufacturer:** The manufacturer of the meter
- **Physical Meter Type:** The type of physical meter
- **Model Number:** The manufacturer's model number for the meter
- **Notes:** Notes about the physical meter

## Physical Meter Location History Table

Records in the Physical Meter Location History table indicate the where and when a specific physical meter is located at a given time. Records in this table include the following information:

- **Physical Meter:** The physical meter associated with the record
- **Location:** The location of the physical meter for this record
- **Arrival Time:** The date and time at which the physical meter arrived at the location specified in this record
- **Reference Number:** A reference number for the physical meter
- **Departure Time:** The date and time at which the physical meter departed from the location specified in this record
- **Storage Location:** Where the physical meter is stored in the location specified in this record
- **Status:** The status of the physical meter, from the Physical Meter Statuses table
- **Notes:** Notes about the physical meter and its storage at the location specified in this record

## Setting Up Meter Data Management Data

Meter Data Management Data includes records which define types of meters and meter events, as well as meter read cycle data. Setting up this data involves creating records in the following tables in the Oracle Utilities Data Repository:

- **MDM Exception Codes Table**
- **Meter Events Table**
- **Meter Event Types Table**
- **Physical Meter Statuses Table**
- **Physical Meter Locations Table**
- **Physical Meter Location Types Table**
- **Physical Meter Types Table**
- **Read Cycle Frequencies Table**
- **Read Cycle Table**
- **Read Cycle Dates Table**

Refer to **Chapter 7: Assets and Setup** in the *Oracle Utilities Meter Data Management User's Guide* for more information about creating these records using the Oracle Utilities Meter Data Management user interface.

### MDM Exception Codes Table

Records in the MDM Exception Codes table represent specific types of exceptions that can occur during processing such as validation and estimation. Records in this table include the following information:

- **UID:** A unique id for the exception code
- **Meter Data Exception Code:** The meter exception code
- **Description:** A description of the exception code
- **Exception Type:** The exception type associated with the exception, from the MDM Exception Types table
- **Exception Category:** The exception category associated with the exception, from the MDM Exception Categories table
- **Work Queue Type:** The Work Queue Type for Work Queue items created when an exception of this type is triggered, from the Work Queue Type table. This overrides the default Work Queue Type defined in the MDM.CFG.XML configuration file.
- **Work Queue:** The Work Queue for Work Queue items created when an exception of this type is triggered, from the Work Queue table. This overrides the default Work Queue defined in the MDM.CFG.XML configuration file.
- **Priority Level:** The priority level assigned to Work Queue items created when an exception of this type is triggered. This overrides the default Priority Level defined in the MDM.CFG.XML configuration file.
- **Assigned to User:** The user ID of the user to whom Work Queue items created when an exception of this type is triggered are assigned. This overrides the default Assigned to User defined in the MDM.CFG.XML configuration file.
- **Automatically Close:** A flag that designates (Yes or No) whether or not Work Queue items created when an exception of this type is triggered are automatically closed when the exception has been resolved.

- **Meter Event:** The Meter Event (from the Meter Events table) associated with the exception code.

The MDM Exception Codes table includes the following predefined records:

<b>Meter Data Exception Code</b>
SCALAR_HIGH_LOW_VALIDATION
SCALAR_GAP_FAILURE
SPIKE_CHECK
INTERVAL_HIGH_LOW_VALIDATION
INTERVAL_SHORT_TERM_GAP_FAILURE
INTERVAL_GAP_FAILURE
ESTIMATION_USAGE
ESTIMATION_FAILURE
ESTIMATION_REQUIRED
ADU TOLERANCE
BROKEN SEAL
TAMPER_ALERT
REVERSE_ROTATION
TILT
RATE_CALL_FAILURE
INACTIVE_METER
ZERO_CONSUMPTION
METER_MULTIPLIER
REPLACEMENT_USAGE
NUMBER_OF_DIALS
NO_HISTORIC_USAGE
SCALAR_HIGH_LOW_NO_HIST_ADU
DELETE_USAGE
READVAL_EXCEEDMAX
VALIDATION_FAILURE
SERIALNUM_INVALID
TLF_INVALID
TOUTOTALIZE_FAILURE
TOUPERIOD_INVALID
TOU_HIGH_LOW

---

**Meter Data Exception Code**

---

INTERVAL\_SUM\_CHECK

---

TOU\_SCHEDULE

---

DEFAULT\_DECIMALS

---

PARTREPLACEMENT\_USAGE

---

OVERLAP\_USAGE

---

KVARH\_FAILURE

---

DEMAND\_HIGH\_LOW\_FAILURE

---

ZERO\_INT\_ENERGY

---

BILLED\_USAGE

---

SEGMENTS\_MERGED

---

SEGMENT\_MISSING

---

POWER\_OUTAGE

---

SP\_SLOWPOLL

---

TP\_TAMPER\_READ\_PENDING

---

DP\_DIAGNOSTIC\_FLAG\_READ\_PENDING

---

TP\_TAMPER\_READ\_COMPLETE

---

IP\_IN\_PROGRESS

---

PE\_PENDING\_EXECUTION

---

RO\_RETRY\_OVERRIDE

---

DS\_DIAGNOSTIC\_FLAG\_READ\_COMPLETE

---

NA|\_DATA\_NOT\_AVAILABLE

---

COMMENT\_CODE

---

TESTMODE\_FAILURE

---

OUTAGE

---

METER\_NOT\_READ

---

BROKEN\_SEAL

---

METER\_HEALTH

---

MISSING\_EX\_STATUS\_MAPPING

---

OUTOFORDER\_ADJUSTMENT

---

NOT\_VALIDATED

---

## Meter Events Table

Records in the Meter Events table represent specific meter-related events. Records in this table include the following information:

- **UID:** A unique id for the meter event
- **Meter Event:** The name of the meter event
- **Meter Event Type:** The meter event type associated with the meter event, from the Meter Event Types table
- **Meter Event Sub Type:** A secondary meter event type associated with the meter event, from the Meter Event Types table
- **Work Queue Type:** The Work Queue Type for Work Queue items created when an event of this type is created, from the Work Queue Type table. This overrides the default Work Queue Type defined in the MDM.CFG.XML configuration file.
- **Work Queue:** The Work Queue for Work Queue items created when an event of this type is created, from the Work Queue table. This overrides the default Work Queue defined in the MDM.CFG.XML configuration file.
- **Priority Level:** The priority level assigned to Work Queue items created when an event of this type is created. This overrides the default Priority Level defined in the MDM.CFG.XML configuration file.
- **Assigned to User:** The user ID of the user to whom Work Queue items created when an event of this type is created are assigned. This overrides the default Assigned to User defined in the MDM.CFG.XML configuration file.
- **Description:** A description of the meter event

The Meter Event table includes the following predefined records:

Meter Event	Meter Event Type	Description
OUTAGE	METERPING	Outage
INFERRED	METERPING	Inferred
NORESPONSE	METERPING	No Response
RESTORATION	METERPING	Restoration
TWACS_UNKNOWN	METERPING	Twacs Unknown

## Meter Event Types Table

Records in the Meter Event Types table represent specific types of meter-related events. Records in this table include the following information:

- **UID:** A unique id for the meter event type
- **Event Type Code:** A code for the event type
- **Description:** A description of the meter event type

The Meter Event Types table includes the following predefined records:

Event Type Code	Description
OUTAGE	Outage



Event Type Code	Description
METERPING	Events received from a meter ping request
MDM_CHAN_EVENT	Logical Meter Event Code
MDM_PHYS_EVENT	Physical Meter Event Code

## Physical Meter Statuses Table

Records in the Physical Meter Statuses table represent various statuses for physical meters, such as New, Repaired, Refurbished, etc. Records in this table include the following information:

- **UID:** A unique id for the physical meter status
- **Status:** The name of the physical meter status

## Physical Meter Locations Table

Records in the Physical Meter Location table represent specific locations where physical meters might be stored or located, such as a warehouse, a repair shop, etc. Records in this table include the following information:

- **UID:** A unique id for the physical meter location
- **Name:** The name of the physical meter location
- **Description:** A description of the location
- **Location Type:** The type of location, from the Physical Meter Location Type table
- **Address:** The address of the location, from the LS Addresses table

## Physical Meter Location Types Table

Records in the Physical Meter Location Type table represent various types of locations where physical meters are located, such as warehouses, stores, or repair shops. Records in this table include the following information:

- **UID:** A unique id for the physical meter location type
- **Location:** The name of the location type

## Physical Meter Types Table

Records in the Physical Meter Type table represent specific types of physical meters. Records in this table include the following information:

- **UID:** A unique id for the physical meter type
- **Physical Meter Type:** The name of the physical meter type
- **Description:** A description of the meter type

## Read Cycle Frequencies Table

Records in the Read Cycle Frequencies table represent the frequencies (weekly, monthly, etc.) at which meters within a given read cycle are read. Records in this table include the following information:

- **Frequency Code:** A code for the read cycle frequency
- **Description:** A description of the read cycle frequency

## Read Cycle Table

Records in the Read Cycle table represent meter read cycles. Meter read cycles determine when specific meters are read. Records in this table include the following information:

- **Read Cycle Name:** The name of the read cycle
- **Description:** A description of the read cycle
- **Frequency Code:** The Read Cycle Frequency code associated with the read cycle, from the Read Cycle Frequency table
- **Start Time:** The start time of the read cycle. This is the date and time after which the read cycle is in effect
- **Stop Time:** The stop time of the read cycle. This is the date and time before which the read cycle is in effect

## Read Cycle Dates Table

Records in the Read Cycle Date table specify the date on which meters associated with a meter read cycle are read. Records in this table include the following information:

- **UID:** A unique ID for the read cycle date record
- **Read Cycle:** The read cycle to which the read cycle date applies
- **Read Date:** The date on which meters associated with read cycle are read

## Setting Up Meter Data Management Validations and Meters

Before usage can be imported into the Oracle Utilities Meter Data Management, the validations to be applied to incoming usage data must be defined in the Oracle Utilities Data Repository. In addition, individual meters must be defined and configured. Creating and configuring meters and defining validations involves creating records in the following tables in the Oracle Utilities Data Repository:

- **Validation Groups Table**
- **MDM Meter Table**
- **Meter Configuration Table**

Refer to **Chapter 3: Meter Management** in the *Oracle Utilities Meter Data Management User's Guide* for more information about creating these records using the Oracle Utilities Meter Data Management user interface.

### Validation Groups Table

Records in the Validation Groups table represent groupings of validations and operations that are applied to meter readings in the Oracle Utilities Meter Data Management application. Records in this table include the following information:

- **UID:** A unique id for the validation group record
- **Name:** The name of the validation group
- **Description:** A description of the validation group
- **Validation Groups:** The specific validations in the validation group. See **Viewing Validations** on page 3-17 in the *Oracle Utilities Meter Data Management User's Guide* for information about viewing individual validations within a validation group. See **Adding, Editing, and Deleting Validations** on page 3-18 in the *Oracle Utilities Meter Data Management User's Guide* for information about editing validations within a validation group.

### MDM Meter Table

Records in the MDM Meter table represent logical meters used to record interval usage and consumption.

#### A Note About Physical and Logical Meters

Physical meters are the devices used to measure usage. Logical meters are used to distinguish between the different types of usage recorded by a single physical meter. More than one logical meter can be associated to a single physical meter. For example, a physical meter might be used to measure multiple registers of usage data. To represent this in Oracle Utilities Meter Data Management, you would create a single Physical Meter record for the actual physical device, and two MDM Meter records: one to record interval usage, the other to record consumption.

Records in this table include the following information:

- **UID:** A unique id for the meter record
- **Meter ID:** The meter ID of the meter
- **Expected Unit of Measure:** The expected unit of measure for the meter, from the **Unit of Measure** table
- **Channel ID:** The channel number associated with the meter
- **Service Point:** The service point associated with the meter

- **End Use:** The end use code associated with the meter, from the **End Usage** table
- **Usage Type Code:** The usage type (CONSUMPTION, INTERVAL, or TOU) associated with the meter, from the **Usage Types** table
- **Related Data Channel:** A related meter, from the **MDM Meter** table
- **Notes:** Notes about the meter
- **Associated KVARH Data Channel:** A related KVARH meter, from the **MDM Meter** table

## Meter Configuration Table

Records in the Meter Configuration table define specific characteristics of a meter and its use within the Oracle Utilities Meter Data Management application. Characteristics defined by meter configuration records include read cycles, time of use schedules, validation groups, profile meters, meter multipliers, number of dials, meter offset, seconds per interval, and other characteristics. Records in this table include the following information:

- **UID:** A unique id for the meter configuration record
- **Meter Data Channel:** The meter corresponding to the meter configuration record, from the MDM Meter table.
- **Physical Meter:** The physical meter corresponding to the meter configuration record, from the Physical Meters table
- **Start Time:** The start time of the meter configuration record.
- **Stop Time:** The stop time of the meter configuration record.
- **Read Cycle:** The read cycle for the meter configuration record, from the Read Cycle table.
- **TOU Schedule:** The time-of-use (TOU) schedule used with the meter configuration record, from the TOU Schedules table.
- **Validation Group:** The validation group used with the meter configuration record, from the Validation Groups table. See **Validation Groups Table** on page 4-17 for more information about validation groups.
- **Profile Meter:** The profile meter associated with the meter configuration record, from the MDM Meter table.
- **Meter Multiplier:** The meter multiplier of the meter defined in the meter configuration record.
- **Number of Dials:** The number of dials on the meter defined in the meter configuration record.
- **Meter Offset:** The meter offset of the meter defined in the meter configuration record.
- **Seconds Per Interval:** The seconds-per-interval (SPI) of the meter defined in the meter configuration record.
- **Default Number of Decimals:** The default number of decimals included in meter readings from the meter defined in the meter configuration record.
- **Pulse Multiplier:** The pulse multiplier of the meter defined in the meter configuration record.
- **Power Flow Direction:** The power flow direction (Received or Distributed) of the meter defined in the meter configuration record. Unless specifically supplied in a meter reading, the Power Flow Direction (DC\_FLOW) of all readings for the meter defined in this record will be set to this setting.

- **Totalize:** A flag that indicates the method (Include, Total, or Exclude) by which the data (interval or scalar) recorded by the meter defined in the meter configuration record is totalized.
- **Billed?:** A flag (Yes or No) that indicates whether or not the data stored by the meter defined in the meter configuration record is required for billing.
- **KE:** The solid state meter constant of the meter defined in the meter configuration record (data schema specific).
- **CT:** The current transformer ratio of the meter defined in the meter configuration record (data schema specific).
- **VT:** The voltage transformer ratio of the meter defined in the meter configuration record (data schema specific).
- **KH:** The watt-hour meter constant of the meter defined in the meter configuration record (data schema specific).
- **TR:** Additional dial constant (for universal register meters) of the meter defined in the meter configuration record (data schema specific).
- **Account ID:** The account associated with the meter defined in the meter configuration record, from the Accounts table
- **Transformer Loss Factor:** The transformer loss factor of the meter defined in the meter configuration record

## Setting Up Oracle Utilities Meter Data Management Bill Determinant Calculation Data

Oracle Utilities Meter Data Management Bill Determinant Calculation Data includes records that define how billing determinants are calculated, including the entities (accounts, service points, and meters) for which billing determinants can be calculated, the specific billing cycles and rate schedules used in those calculations and other data. Setting up this data involves creating records in the following tables in the Oracle Utilities Data Repository:

- **Account Table**
- **Account History Table**
- **BD Queue Type Table**
- **BD Queue Table**
- **BD Queue Archive Table**
- **BD Request Table**
- **Bill Determinants Table**
- **Bill History Table**
- **Bill History Values Table**
- **Bill View Table**
- **Billing Cycle Table**
- **Billing Cycle Dates Table**
- **Customer Table**
- **Meter Bill Determinant Values Table**
- **Rate Code Table**
- **Rate Code History Table**
- **Rate Form Table**
- **Rate Form Version Table**
- **Service Point Table**
- **Service Point Accounts Table**
- **Service Point Bill Determinant Values Table**

Table designated are run time tables are populated during the billing determinant calculation process, and do not require set up or configuration.

**Note:** Many of the tables used with Oracle Utilities Meter Data Management billing determinant calculations are also used by Oracle Utilities Billing Component. If both Oracle Utilities Meter Data Management and Oracle Utilities Billing Component are licensed, it is recommended that only one of the products be used for all billing calculations.

## Account Table

Records in the Account table represent accounts for which billing determinants are calculated. Accounts are linked to customers in the **Customer** table, and to service points in the **Service Point Accounts** table. Records in this table include the following information:

- **UID:** The unique ID of the account.
- **Customer:** The parent customer ID, from the **Customer Table** table.
- **Account ID:** The account ID.
- **Start Time:** Start time for the account.
- **Stop Time:** Stop time for the account. Accounts with a non-Null stop date are no longer active accounts. Billing determinants cannot be generated for inactive accounts.
- **Operating Company:** Operating company associated with the account.
- **Jurisdiction:** Jurisdiction associated with the account.
- **Name:** The name of the account.
- **SIC:** The Standard Industrial Classification (SIC) code for the account.
- **Account Status:** The current status of the account.
- **Revenue Code:** Revenue code for the account.
- **Billing Mode:** Billing mode for the account. When setting up accounts for meter data management bill determinant calculations, this must be set to **Fully Automatic**.
- **Print Detail:** Specifies the level of information provided on billing reports and transaction records. This is not applicable to bill determinant calculations.
- **Full Day Bill:** A flag that specifies how the end of a billing period is recorded. This can be one of the following:
  - **Yes:** Use midnight of the last full day of the billing period.
  - **No:** Use the actual read date and time.
  - **Use Default:** Use the flag specified in either the rate code or rate schedule (in that order). If neither is available, use the actual read date and time.
- **Pre Window:** The number of days prior to the scheduled read date when an account becomes eligible for billing determinant calculations. This can be any value from 0 to 28.
- **Post Window:** The number of days after the scheduled read date when an account remains eligible for billing determinant calculations. This can be any value from 0 to 28.
- **EDI:** A flag that indicates if the account supports EDI format.
- **Region:** Region for the account.
- **LS Currency:** The currency code for the account, from the LSCurrency table.
- **Owner:** The owner of the account.
- **LS Account Type:** The account type.
- **Market Participant ID:** ID of the market participant associated with the account.
- **Organization:** Organization associated with the account.
- **Industrial Classification Code:** The Industrial Classification code for the account.
- **Special Handling Code:** Handling code for the account.

## Account History Table

Records in the Account History table store relationships between accounts and billing cycles (defined in the **Billing Cycle** table). These records help Oracle Utilities Meter Data Management determine the correct read (bill) date for an account. Records in this table include the following information:

- **Account:** The account associated with the relationship.
- **Effective Date:** The date on which the relationships defined in the record are effective.
- **Contract:** The Rules Language contract associated with the account.
- **Billing Cycle:** The Billing Cycle (from the **Billing Cycle** table) associated with the account.
- **Bi-Monthly:** A flag that indicates if the account is to be billed b-monthly.
- **Full Day Bill:** A flag that indicates if the account should be billed using Full day bill.
- **Billing Cycle - Governing:** The governing Billing Cycle for the account.
- **Bill on Read Date:** A flag that indicates if the account is to be billed on or before the read date.

## BD Queue Type Table

Records in the BD Queue Type table define specific queues used when initiating billing determinant calculation requests. Queue types allows for separating calculations based on prioritization or other factors. Records in this table include the following information:

- **Business Status Code:** Business status for the queue
- **Description:** A description of the business status

The BD Queue Type table includes the following pre-defined records:

Business Status Code	Description
NORMAL	Standard Priority

## BD Queue Table

Records in the BD Queue table represent individual billing determinant calculation processes (one for each account) that are currently being executed. Each BD Queues record will have a related record in the **BD Request** table. Records in this table include the following information:

- **ID:** ID of the queue record
- **Account:** The account ID associated with the queue record
- **Read Date:** The read date associated with the queue record
- **BD Queue Request:** The ID of the parent BD Request record
- **BD Status Code:** The status code of the queue record (Completed, Error, New, or Processing)
- **BD Queue Type:** The queue type associated with the queue record (from the **BD Queue Type** table)
- **Handler:** The server, port, and service on which the queue record was processed
- **Received Date:** The date on which the request was received
- **Business Status Code:** The business status code (if any) associated with the queue record (from the Business Statuses table).



- **Business Message:** The business message associated with the queue record
- **Completed Date:** The date on which the billing determinant calculations were completed.
- **Calculation Time:** The date on which the queue record was processed.
- **External ID:** The ID of the request when submitted from an external system.

**Note:** This is a run time table.

## BD Queue Archive Table

Records in the BD Queues Archive table represent individual billing determinant calculation processes that have been previously executed. Each BD Queue Archive record will have a related record in the **BD Request** table. Records in this table include the following information:

- **ID:** ID of the queue record
- **Account:** The account ID associated with the queue record
- **Read Date:** The read date associated with the queue record
- **BD Queue Request:** The ID of the parent BD Request record
- **BD Status Code:** The status code of the queue record (Completed, Error, New, or Processing)
- **BD Queue Type:** The queue type associated with the queue record (from the **BD Queue Type** table)
- **Handler:** The server, port, and service on which the queue record was processed
- **Received Date:** The date on which the request was received
- **Business Status Code:** The business status code (if any) associated with the queue record (from the Business Statuses table).
- **Business Message:** The business message associated with the queue record
- **Completed Date:** The date on which the billing determinant calculations were completed.
- **Calculation Time:** The date on which the queue record was processed.
- **External ID:** The ID of the request when submitted from an external system.

**Note:** This is a run time table.

## BD Request Table

Records in the BD Request table represent individual billing determinant calculation requests. Each BD Request record will have one or more related records in the **BD Queue** or **BD Queue Archive** tables. Records in this table include the following information:

- **BD Request ID:** The ID of the request record
- **Description:** A concatenated string comprising the accounts specified for the request (account UID for a single account, list name of account list, or “All”), the Read Date of the submitted request, and the user ID of the user who initiated the request, separated by commas.
- **Date Received:** The date and time on which the request was received
- **Requested User:** The user ID of the user who initiated the request
- **Time Stamp:** The date on which the request was processed

**Note:** This is a run time table.

## Bill Determinants Table

Records in the Bill Determinants table represent types of billing determinants that can be calculated. The actual values for bill determinants are stored in the **Bill History Values**, **Meter Bill Determinant Values**, and **Service Point Bill Determinant Values** tables. Records in this table include the following information:

- **Code:** A unique code (up to 64 characters) for the billing determinant.
- **Identifier:** A unique identifier (up to 32 characters) that the Rules Language will use to refer to the billing determinant.
- **Name:** The complete name or other description (up to 64 characters) of the billing determinant. This name helps users recognize the code in displays and reports.
- **Unit of Measure:** The UOM for this billing determinant (from the **Unit of Measure (UOM)** table).
- **BILLHISTORY Column:** *View only.* If actual usage values for this billing determinant are stored in the Bill History Table, this is the name of the column in the Bill History Table where the values are stored. If the bill determinant values are stored in the Bill History Value Table, this field is blank. This field is only used if additional fields have been added to the Bill History table.
- **Aggregate:** The data aggregation method the programs will use when converting interval data using this Bill Determinant. For example, kW measured every 15 minutes should not be added together when aggregating to hourly data, but should have an average or a maximum taken. The options are:
  - **Average** - average interval values to get aggregate
  - **Max** - find maximum and use as aggregate
  - **Total** - add values to get aggregate

## Bill History Table

Records in the Bill History table represent billing periods for which billing determinants are calculated. The results of billing determinant calculations are stored in the **Bill History Values**, **Meter Bill Determinant Values**, and **Service Point Bill Determinant Values**. Records in this table include the following information:

- **UID:** A unique id for the record
- **Account:** The account associated with the billing period
- **Start Time:** The start time of the billing period
- **Stop Time:** The stop time of the billing period
- **Bill Month:** The bill month of the billing period, designated as MM/YYYY
- **Bill Time:** The time at which billing determinants were calculated for the billing period
- **Read Date:** The bill date (the date on which billing determinants should be calculated) for the billing period
- **User-Specified Stop:** The start time of the billing period
- **KWH:** Energy consumption, measured in KWH, for the billing period for the account.
- **Bill Code:** Bill code associated with the billing period
- **Profile Status:** Profile status associated with the billing period
- **Rebill Reason Code:** Rebill reason code associated the billing period.

- **Published:** Flag indicating if bill determinants calculated for the billing period have been published by the Calculator Engine publisher.

## Bill History Values Table

The Bill History Values is used to store account-level billing determinant calculation results for a billing period in the **Bill History** table. Records in this table include the following information:

- **Bill History:** Identity of the parent Bill History record (Account ID and Start Time, from the **Bill History** table)
- **Bill Determinant:** The billing determinant calculated (from the **Bill Determinants** table)
- **Value:** The calculated value for the billing determinant
- **Reported?:** A flag that indicates (Yes or No) if the billing determinant result has been reported or sent to a down-stream billing system
- **Report Date:** The date on which the billing determinant result was reported or sent to a down-stream billing system

**Note:** This is a run time table.

## Bill View Table

The Bill View table is used store billing determinant calculation results for a billing period. Results are stored in XML format. Records in this table contain the following information:

- **Bill History:** Identity of the parent Bill History record (Account ID and Start Time, from the **Bill History** table)
- **Bill Time:** The time at which billing determinants were calculated for the billing period
- **Bill History Date:** Results of the billing determinant calculations in XML format.

**Note:** This is a run time table.

## Billing Cycle Table

Records in the Billing Cycle table identify the billing cycles used by operating companies/ jurisdictions. The actual dates for each billing cycle are stored in the **Billing Cycle Dates** table. Records in this table include the following information:

- **Operating Company Code:** The operating company that uses this billing cycle.
- **Jurisdiction:** The jurisdiction that uses this billing cycle is applied.
- **Code:** A unique code (up to 64 characters) identifying the billing cycle.

## Billing Cycle Dates Table

Records in the Billing Cycle Dates table define the read date for each bill month in a billing cycle. Records in this table include the following information:

- **Billing Cycle:** The billing cycle to which the dates apply (from the **Billing Cycle** table). This includes the operating company and jurisdiction of the billing cycle.
- **Read Date:** The meter read date, in the MM/DD/YYYY format. This field is used by the system to match an account's Bill History or Meter Value records with the correct bill period.
- **Bill Month:** The bill month, in the MM/YYYY format.

## Business Status Table

Records in the Business Status table define specific statuses that are used to define billing determinant calculation processes (in the **BD Queue** and **BD Queue Archive** tables). Records in this table include the following information:

- **Business Status Code:** Business status code
- **Description:** A description of the business status code

The Business Status table includes the following predefined records:

Business Status Code	Description
DUPLICATE	
ERROR	
FAIL	
PASS	
POSTED	

## Customer Table

Records in the Customer table represent customers which are the parents of accounts (see the **Account Table** on page 4-21 table). Records in this table include the following information:

- **UID:** The unique ID of the customer record in the Oracle Utilities Data Repository (read-only)
- **ID:** The customer ID.
- **Name:** The name of the customer.
- **Summary Bill:** A flag that indicates if Oracle Utilities Billing Component should produce a summary bill for the customer, or bill the customer's account individually.
- **Directory - Account Manager:** The account manager for the customer, from the Directory table.
- **Directory - Contact:** The contact for the customer, from the Directory table.
- **Account Manager Fax Flag:** A flag that indicates if Oracle Utilities Billing Component should send billing results to the Account Manager via email or fax.
- **Contact Fax Flag:** A flag that indicates if Oracle Utilities Billing Component should send billing results to the Contact via email or fax.
- **Credit Score:** A score that represents the payment performance history of the customer (used by the Collections Management Extension).

**Note:** If you have added customized fields to the Customer table in the Oracle Utilities Data Repository, they may also appear on this tab.

- **Owner:** The user ID of the user who last updated the customer record.

## Meter Bill Determinant Values Table

The Meter Bill Determinant Values table is used to store meter-level billing determinant calculation results for a billing period in the **Bill History** table. Records in this table include the following information:

- **UID:** Unique ID for the record
- **Meter Data Channel:** Meter data channel for which the billing determinant value was calculated (from the **MDM Meter Table** table)

- **Bill History:** Identity of the parent Bill History record (Account ID and Start Time, from the **Bill History Table** table)
- **Bill Determinant:** The billing determinant calculated (from the **Bill Determinants** table)
- **Value:** The calculated value for the billing determinant
- **Reported?:** A flag that indicates (Yes or No) if the billing determinant result has been reported or sent to a down-stream billing system
- **Report Date:** The date on which the billing determinant result was reported or sent to a down-stream billing system

**Note:** This is a run time table.

## Rate Code Table

Records in the Rate Code table define codes associated with rate schedules (from the **Rate Form Table** table) that define the business rules used in billing determinant calculations. Records in this table include the following information:

- **Rate Schedule:** The rate schedule associated with the rate code.
- **Code:** A unique code for the rate code.
- **Note:** An optional note for the rate code.
- **Billing Mode Flag:** An optional flag that indicates the billing mode for bills calculated using this rate code. This is not applicable to bill determinant calculations.
- **Print Detail:** An optional flag that indicates the print detail (None, Normal Detail, or All Detail) for bills calculated using this rate code. This is not applicable to bill determinant calculations.
- **Full Day Bill:** An optional flag that indicates if accounts billed using this rate code should be billed using Full Day Bill. This is not applicable to bill determinant calculations.

## Rate Code History Table

Records in the Rate Code History table represent relationships between accounts and rate codes (from the **Rate Code** table). Oracle Utilities Meter Data Management uses the records in this table to determine which rate schedules to use when processing billing determinant calculations for an account. Records in this table include the following information:

- **Account:** The account associated with the rate code.
- **Start Time:** The start time for the relationship between the account and the rate code.
- **Rate Code:** The rate code associated with the account (from the **Rate Code** table). This includes the operating company, jurisdiction, rate schedule code, and rate code.
- **Stop Time:** *Optional.* The stop time for the relationship between the account and the rate code.
- **Note:** An optional note for the rate code.

## Rate Form Table

Records in the Rate Form table represent rate schedules that define the business rules used in billing determinant calculations. See **Rate Forms** on page 1-4 in the *Oracle Utilities Rules Language User's Guide* for more information about rate form versions. Records in this table include the following information:

- **UID:** A unique id for the rate form
- **Operating Company:** Operating company associated with the rate form

- **Jurisdiction:** Jurisdiction associated with the rate form
- **Code:** A code used to designate the rate form
- **Type:** The type of rate form. When defining rate forms for use with Oracle Utilities Meter Data Management, valid types include Schedule and Rider.
- **Name:** The name of the rate form
- **Note:** A note about the rate form
- **Billing Mode Flag:** An optional flag that indicates the billing mode for bills calculated using this rate form. This is not applicable to bill determinant calculations.
- **Print Detail:** An optional flag that indicates the print detail (None, Normal Detail, or All Detail) for bills calculated using this rate form. This is not applicable to bill determinant calculations.
- **Full Day Bill:** An optional flag that indicates if accounts billed using this rate form should be billed using Full day bill. This is not applicable to bill determinant calculations.
- **Editable:** A flag that indicates if the rate form is editable. This should always be set to No when creating new rate form records.

## Rate Form Version Table

Records in the Rate Form Version table represent different versions of a rate form defined in the **Rate Form** table. See **Rate Form Versions** on page 1-5 in the *Oracle Utilities Rules Language User's Guide* for more information about rate form versions. Records in this table include the following information:

- **UID:** A unique id for the rate form version
- **Rate Form:** The parent rate form for the version (from the **Rate Form** table).
- **Version No.:** The version number of the rate form version. Applicable only to Trial versions.
- **Effective Date:** The date on which the rate form version is in effect. Applicable only to Current and Historical versions.

**Note:** Rate Form Version records should be created using Data Manager via the File->New->Rate Form Version menu option. See **The Rules Language Editor** on page 2-2 in the *Oracle Utilities Rules Language User's Guide* for more information.

## Service Point Table

Service points are points where service is provided. Bill determinants can be calculated for each service point associated with an account. Service points are linked to accounts in the **Service Point Accounts** table and are linked to meters in the **MDM Meters** table. Records in this table include the following information:

- **UID:** A unique id for the service point
- **Market ID:** Market ID associated with the service point
- **Service Point ID:** ID for the service point
- **Service Point Name:** The name of the service point
- **Premise:** The premise (from the LS Premise table) associated with the service point.
- **Address:** The address of the service point (from the LS Address table)
- **Service Point Class:** The class of the service point (from the Service Point Class table)
- **Special Needs?:** A flag that designates (Yes or No) if the service point has special needs
- **Eligibility Date:** The date on which the service point is eligible for service

- **Transformer Location:** The transformer location associated with the service point (from the Transformer Location table)
- **Start Date:** The start date of the service point
- **Stop Date:** The stop date of the service point

## Service Point Accounts Table

Records in the Service Point Accounts represent relationships between accounts (from the **Account** table) and service points (from the **Service Point** table). Records in this table include the following information:

- **Service Point ID:** ID for the service point
- **Account:** ID for the account
- **Start Time:** The start date and time of the relationship
- **Stop Time:** The stop date and time of the relationship

## Service Point Bill Determinant Values Table

The Service Point Bill Determinant Values table is used to store service point-level billing determinant calculation results for a billing period in the **Bill History** table. Records in this table include the following information:

- **UID:** Unique ID for the record
- **Service Point:** The service point for which the billing determinant value was calculated (from the **Service Point Table** table)
- **Bill History:** Identity of the parent Bill History record (Account ID and Start Time, from the **Bill History Table** table)
- **Bill Determinant:** The billing determinant calculated (from the **Bill Determinants** table)
- **Value:** The calculated value for the billing determinant
- **Reported?:** A flag that indicates (Yes or No) if the billing determinant result has been reported or sent to a down-stream billing system
- **Report Date:** The date on which the billing determinant result was reported or sent to a down-stream billing system

**Note:** This is a run time table.



## Meter Data Management Work Queues

Oracle Utilities Meter Data Management uses the Work Queues component of the Oracle Utilities Energy Information Platform for exception management during validation processing. A set of Work Queues related records is included in the Oracle Utilities Meter Data Management database schema.

### A Note about Predefined Records:

The Work Queue records described in this section are used by the Oracle Utilities Meter Data Management application. **Do not modify or delete any of these predefined records.** Changes made to these records can result in errors when using the Oracle Utilities Meter Data Management application.

## Work Queue

The Work Queue table includes the following predefined records:

Code	Description
MDM	Meter Data Management
MDMBD	Meter Data Management Billing Determinants

The MDM work queue is used as the default queue for work queue items generated by the Oracle Utilities Meter Data Management application.

## Work Queue Type

The Work Queue Type table includes the following predefined records:

Work Queue Type	Description	Default Work Queue
CRITICAL	MDM Critical Data Error	MDM
MDMBD	BDQUEUE Error	MDMBD
MDMBD_INSUF_USAGE	Insufficient Usage Data for this Account	MDMBD
MDMBD_NO_BH	No Historical Bill History Records were found for this account	MDMBD
MDMBD_NO_RC	No Rate Codes were found for this account	MDMBD
USAGE	MDM Usage Error	MDM

## Work Queue Hints

The Work Queue Hint table includes the following predefined records:

Name	Description
Adapter Re-Submit	Adapter Re-Submit
Adapter Runtime	See Adapter Runtime Exception
Associate Rate Code	Associate Rate Code Record to Account

Name	Description
Create BH Record	Create Bill History Record for Account
MDM Exception	Usage Exception
Meter Search	Meter Search
Review Usage	Review Usage for this Account for the BD Period
Usage Error	Usage Error

These hints allow users to perform the following actions:

- **Adapter Re-Submit:** Re-submits a failed usage reading for validation via the Oracle Utilities Adapter
- **Adapter Runtime:** Opens the Oracle Utilities Adapter Inbound screen, for the usage record that triggered the exception
- **Associated Rate Code:** Opens the Account dashboard for the account. Click Account-Related Tables on the Account Basics clip to add a Rate Code History record for the account.
- **Create BH Record:** Opens the Account dashboard for the account. Use the Account Billing clip to create a Bill History record for the account and billing period.
- **MDM Exception:** Opens the Exceptions tab on the Edit Values screen for the usage reading that triggered the exception
- **Meter Search:** Opens the Usage Add/Edit search screen
- **Review Usage:** Opens the Account dashboard for the account. Use the Meter Data Management clip to view usage for the account.
- **Usage Error:** Opens the Exceptions tab on the Edit Values screen for the usage reading that triggered the exception

## Work Queue Type Hint

The Work Queue Type Hint table includes the following predefined records:

Work Queue Type	Work Queue Hint Name
CRITICAL	Adapter Re-Submit
CRITICAL	Adapter Runtime
CRITICAL	Meter Search
MDMBD_INSUF_USAGE	Review Usage
MDMBD_NO_BH	Create BH Record
MDMBD_NO_RC	Associate Rate Code
USAGE	Usage Error

## Creating Custom Work Queue Records for use with Oracle Utilities Meter Data Management

By default, the Oracle Utilities Meter Data Management application uses the Work Queue Type, Work Queue, and Work Queue Hints described above for all exceptions generated during Oracle Utilities Meter Data Management processing. However, you can create custom work queue types and queues as part of your Oracle Utilities Meter Data Management implementation.

For example, you might want work queue items generated from specific Oracle Utilities Meter Data Management exceptions to go to a specific work queue rather than to the default “MDM” work queue. In addition, you might want to create specific work queue types for specific types of Oracle Utilities Meter Data Management exceptions.

### Create Custom Work Queue and Work Queue Type

The first step is to create records in the Work Queue and/or Work Queue Type tables for your custom work queues.

For example, if you wanted all work queue items generated based on estimation-related MDM Exceptions to go to an “Estimation” work queue, you might create a record in the Work Queue table as follows:

- **Work Queue:** ESTIMATION
- **Description:** Estimation Work Queue Items

If you wanted the “ESTIMATION\_FAILURE” MDM Exception to generate work queue items of type “Estimation Failure,” you could create a record in the Work Queue Type table as follows:

- **Work Queue Type:** Estimation Failure
- **Description:** Failed Estimation
- **Work Queue:** ESTIMATION
- Etc.

Refer to **Defining Your Work Queues Configuration** on page 5-8 and **Configuring Work Queues and Work Queue Types** on page 5-11 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about creating custom work queues and work queue types.

## Add Custom Work Queue/Work Queue Type to MDM Exceptions

The next step is to associate your custom work queues and/or work queue types with the MDM Exception Code that you wish to use to create your custom work queues.

For example, to associate your “ESTIMATION” work queue with the estimation-related MDM Exceptions (ESTIMATION\_USAGE, ESTIMATION\_FAILURE, and ESTIMATION\_REQUIRED) you would set the Work Queue column of the MDM Exception Code records for the above exception types to “ESTIMATION.”

To associated your “Estimation Failure” work queue type to the “ESTIMATION\_FAILURE” MDM Exception Code, you would set the Work Queue Type column of the “ESTIMATION\_FAILURE” MDM Exception Code record to “Estimation Failure.”

See **Meter Exception Codes** on page 7-16 in the *Oracle Utilities Meter Data Management User's Guide* for more information about updating MDM Exception Codes using the Oracle Utilities Meter Data Management user interface.

## A Note About Oracle Utilities Meter Data Management Work Queue Hints

The Work Queue Hints described in this chapter were designed specifically to work with the pre-defined Work Queue, Work Queue Type, and MDM Exception Codes provided with the Oracle Utilities Meter Data Management application. These Work Queue Hints may not work properly with custom work queue types.

# Chapter 5

---

## Setting Up Oracle Utilities Meter Data Management Processes

This chapter describes how to set up processes used by Oracle Utilities Meter Data Management, including:

- **Oracle Utilities Meter Data (.LSM) File Format**
- **Oracle Utilities Meter Data Management Business Rule Types**
- **Oracle Utilities Meter Data Management Data Converters**
- **Configuring Usage Data Import using the Adapter**
- **Oracle Utilities Meter Data Management Validations and Estimations**

## Oracle Utilities Meter Data (.LSM) File Format

Oracle Utilities Meter Data Management uses the Oracle Utilities Meter Data (.LSM) format for usage readings. This section describes this format, including:

- **LSM Format Examples**
- **LSM Format Element Descriptions**
- **LSM Format Schema**

### LSM Format Examples

The Oracle Utilities Meter Data (LSM) format can be used for consumption, time of use, and interval usage readings. In addition, usage in the LSM format can be provided based on logical or physical meters. Examples of each are provided below.

#### Consumption Example

```
<LODESTARUSAGE>
  <MDMUSAGE METERID="ASTROS" CHANNELID="" UOMCODE="01">
    <METERREADS>
      <METERREAD STARTREADTIME="2005-10-19T19:30:00" STOPREADTIME="2005-10-
19T20:30:00" STARTVAL="1000" STOPVAL="3000" TOTALVAL="2000" MINVAL="" MAXVAL=""
ENERGY="2000" METERMULTIPLIER="" NUMOFDIALS="" METEROFFSET=""
PULSEMULTIPLIER="" TOTALIZE="" DCFLOW="" TOUPERIODNAME="" />
    </METERREADS>
  </MDMUSAGE>
</LODESTARUSAGE>
```

#### Time Of Use (TOU) Example

```
<LODESTARUSAGE>
  <MDMUSAGE METERID="ASTROS" CHANNELID="" UOMCODE="01">
    <METERREADS>
      <METERREAD STARTREADTIME="2005-10-19T19:30:00" STOPREADTIME="2005-10-
19T20:30:00">
        <TOUPERIODS>
          <TOUPERIOD TOUPERIODNAME="ONPEAK" STARTVAL="1000" STOPVAL="3000"
TOTALVAL="2000" MINVAL="" MAXVAL="" ENERGY="2000" METERMULTIPLIER=""
NUMOFDIALS="" METEROFFSET="" PULSEMULTIPLIER="" PULSEOFFSET="" TOTALIZE=""
DCFLOW="" />
        </TOUPERIODS>
      </METERREAD>
    </METERREADS>
    <METEREVENTS>
      <METEREVENT EVENTCODE="" LOGDATE="" NOTES="" />
    </METEREVENTS>
  </MDMUSAGE>
</LODESTARUSAGE>
```

## Interval Example

```

<LODESTARUSAGE>
  <MDMUSAGE METERID="ASTROS" CHANNELID="26" UOMCODE="01">
    <METERREADS>
      <METERREAD STARTREADTIME="2005-10-19T19:30:00" STOPREADTIME="2005-10-
19T20:29:59" STARTVAL="1000" STOPVAL="3000" TOTALVAL="" ENERGY="" SPI="900"
DESCRIPTOR="WORLD SERIES" METERMULTIPLIER="" NUMOFDIALS="" METEROFFSET=""
PULSEMULTIPLIER="" PULSEOFFSET="" TOTALIZE="" DCFLOW="" DSTPARTICIPANT=""
TRANSLATIONTIME="" TZSTDNAME="" ACCEPTREJECTSTATUS="" POPULATION="" WEIGHT=""
ORIGIN="" POWERFACTOR="">
        <INTS>
          <I V="50" S="" />
          <I V="70" S="" />
          <I V="60" S="" />
          <I V="80" S="" />
        </INTS>
        <CUSTOM/>
      </METERREAD>
    </METERREADS>
    <METEREVENTS>
      <METEREVENT EVENTCODE="" LOGDATE="" NOTES="" />
    </METEREVENTS>
  </MDMUSAGE>
</LODESTARUSAGE>

```

## Interval Example - Physical Meter

```

<LODESTARUSAGE>
  <MDMUSAGE HEADERSERIALNUMBER="12345" METERMANUFACTURER="METERSRUS"
EFFECTIVEDATE="2005-01-01T00:00:00">
    <METERREADS>
      <METERREAD STARTREADTIME="2005-10-19T19:30:00" STOPREADTIME="2005-10-
19T20:29:59" STARTVAL="1000" STOPVAL="3000" TOTALVAL="" ENERGY="" SPI="900"
DESCRIPTOR="WORLD SERIES" METERMULTIPLIER="" NUMOFDIALS="" METEROFFSET=""
PULSEMULTIPLIER="" PULSEOFFSET="" TOTALIZE="" DCFLOW="" DSTPARTICIPANT=""
TRANSLATIONTIME="" TZSTDNAME="" ACCEPTREJECTSTATUS="" POPULATION="" WEIGHT=""
ORIGIN="" POWERFACTOR="">
        <INTS>
          <I V="50" S="" />
          <I V="70" S="" />
          <I V="60" S="" />
          <I V="80" S="" />
        </INTS>
        <CUSTOM/>
      </METERREAD>
    </METERREADS>
    <METEREVENTS>
      <METEREVENT EVENTCODE="" LOGDATE="" NOTES="" />
    </METEREVENTS>
  </MDMUSAGE>
</LODESTARUSAGE>

```

## LSM Format Element Descriptions

In the descriptions below, the type of meter to which the element or attribute applies is listed in parentheses as follows:

- A - applies to ALL meters
- C - applies to consumption meter
- I - applies to INTERVAL meters
- T - applies to Time of Use (TOU) meters

**LODESTARUSAGE:** Root element that contains one or more MDMUSAGE elements

**MDMUSAGE:** Element that contains one or more METERREADS and/or METEREVENTS elements

Attributes:

**METERID:** Meter ID (A)

**CHANNELID:** Channel ID (A)

**UOMCODE:** Unit of measure code (A)

**HEADERSERIALNUMBER:** The serial number of the physical meter from which the meter reading comes. Used conjunction with the METERMANUFACTURER and EFFECTIVEDATE to lookup the METERID (logical meter) for the meter when not supplied in the reading. (A)

**METERMANUFACTURER:** The meter manufacturer of the physical meter from which the meter reading comes. Used in conjunction with the HEADERSERIALNUMBER and EFFECTIVEDATE to lookup the METERID (logical meter) for the meter when not supplied in the reading. (A)

**EFFECTIVEDATE:** The date at which time the physical meter is active (i.e. associated with a logical meter). Used in conjunction with the HEADERSERIALNUMBER and METERMANUFACTURER to lookup the METERID (logical meter) for the meter when not supplied in the reading.(A)

Elements:

**METERREADS:** Element containing one or more METERREAD elements

Elements:

**METERREAD:** Element that defines a single meter reading

Attributes:

**STARTREADTIME:** Start time of the meter reading (A)

**STOPREADTIME:** Stop time of the meter reading (A)

**STARVAL:** Start value of the meter reading (A)

**STOPVAL:** Stop value of the meter reading (A)

**TOTALVAL:** Total value of the meter reading (A)

**MINVAL:** Minimum value of the meter reading (A)

**MAXVAL:** maximum value of the meter reading (A)

**ENERGY:** Energy value of the meter reading (A)

**DESCRIPTOR:** Description of the meter reading (I)

**SPI:** The duration of each interval (measure in seconds-per-interval) in the meter reading. Applies only to readings whose USAGETYPECODE is INTERVAL (I)

**NUMOFDIALS:** Number of dials for the meter reading (C, T)

**METERMULTIPLIER:** Meter multiplier for the meter reading (A)

**METEROFFSET:** Meter offset for the meter reading (A)

**PULSEMULTIPLIER:** Pulse multiplier for the meter reading (A)

**PULSEOFFSET:** Pulse offset for the meter reading (A)

**POWERFACTOR:** Power factor for the meter reading (I)



**CHNSTATUS:** The integer value of the extended status code for the channel. Valid integer values start from 1. The channel status code displayed on the user interface is based on the EXTENDEDSTATUSES.CFG.XML configuration file. (I)

To specify multiple channel status codes, this should be set to an integer that is equal to the sum of the integer values the correspond to the status codes you wish to specify. For example, to specify status codes “NA” (1) and “AD” (2), you would set this equal to “3”. To specify status codes “NA” (1), “AD” (2), and “ES” (8), you would set this equal to “11”.

**DSTPARTICIPANT:** Indicating whether or not this record will be processed using DST adjustments. Valid values are Y, N, or A. Applies only to readings whose USAGETYPECODE is INTERVAL

**TOTALIZE:** Indicates the method (INCLUDE, TOTAL, or EXCLUDE) by which the data recorded by the meter reading is totalized.

**DCFLOW:** The power flow direction (RECEIVED or DISTRIBUTED) of the meter defined in the meter configuration record.

**ORIGIN:** Origins of meter reading. Must be one of: M (metered), P (profiled), C (computed), or S (Oracle Utilities Load Analysis Statistic). Applies only to readings whose USAGETYPECODE is INTERVAL (I)

**WEIGHT:** Used by Oracle Utilities Load Analysis Statistical records only (I)

**POPULATION:** Used by Oracle Utilities Load Analysis Statistical records only (I)

**ACCEPTREJECTSTATUS:** Used by MV90 to indicate if the reading was accepted or rejected. Applies only to readings whose USAGETYPECODE is INTERVAL (I)

**TRANSLATIONTIME:** Used by MV90 to indicate the date/time the reading was translated from MV90. Applies only to readings whose USAGETYPECODE is INTERVAL (I)

**TZSTDNAME:** Time zone standard name for the meter reading. Applies only to readings whose USAGETYPECODE is INTERVAL (I)

**READDATE:** Read date for the meter reading (A)

**READMETHOD:** Read method for the meter reading (A)

**SERIALNUMBER:** Serial number of the physical meter from which the meter reading comes. Used to lookup the METERID for the meter when not supplied in the reading. (A)

**TRANSFORMERLOSSFACTOR:** Transformer loss factor for the meter reading (A)

**DEFAULTDECIMALS:** Default number of decimal places attribute for the meter reading (A)

Elements:

**TOUPERIODS:** Element that one or more TOUPERIOD elements.

Elements:

**TOUPERIOD:** Element that defines a time of use (TOU) reading.

Attributes:

**TOUPERIODNAME:** Name of the time of use period associated to the meter reading (I)

**STARTVAL:** Start value of the meter reading (I)

**STOPVAL:** Stop value of the meter reading (I)

**TOTALVAL:** Total value of the meter reading (I)

**ENERGY:** Energy value of the meter reading (I)

**METERMULTIPLIER:** Meter multiplier for the meter reading (I)

**NUMOFDIALS:** Number of dials for the meter reading (I)

**INTS:** Element that contains one or more I elements, each of which define an individual interval reading

Elements:

**I:** Element that defines a single interval reading

Attributes:

**V:** The value of the interval (I)

**S:** The status code of the interval (I)

**E:** The integer value of the extended status code for the interval. Valid integer values start from 1. The interval status code(s) displayed on the user interface are based on the Codes and Descriptions defined in the EXTENDEDSTATUSES.CFG.XML configuration file. (I)

To specify multiple extended status codes, this should be set to an integer that is equal to the sum of the integer values the correspond to the status codes. For example, to specify status codes “PO” (1) and “SI” (2), you would set this equal to “3”. To specify status codes “PO” (1), “SI” (2), and “CR” (8), you would set this equal to “11”.

**TS:** Timestamp of the interval (I). Not currently supported.

**CUSTOM:** Element that defines custom attributes of the meter reading

Attributes:

**METEREVENTS:** Element containing one or more METEREVENT elements.

Elements:

**METEREVENT:** Element defining a single meter event

Attributes:

**STARTREADTIME:** Start time of the meter reading (A)

**STOPREADTIME:** Stop time of the meter reading (A)

**EVENTCODE:** Event code for the meter event (A)

**LOGDATE:** The date on which the meter event is to be logged (A)

**NOTES:** Notes concerning the meter event (A)

**MDMEVENTS:** Element containing one or more MDMEVENT elements. Used to define Physical Meter Events.

Attributes:

**SERIALNUMBER:** Required. The serial number of the physical meter for the meter event. Used conjunction with the MANUFACTURER to lookup the METERID (logical meter) for the meter when not supplied in the reading. (A)

**MANUFACTURER:** Optional. The meter manufacturer of the physical meter for the meter event. Used in conjunction with the SERIALNUMBER to lookup the METERID (logical meter) for the meter when not supplied in the reading. (A)

Elements:

**MDMEVENT:** Element defining a single physical meter event

Attributes:

**EVENTCODE:** Event code for the physical meter event (A)

**LOGDATE:** The date on which the physical meter event is to be logged (A)

**NOTES:** Notes concerning the physical meter event (A)

## LSM Format Schema

Below is the XML schema for the LSM format.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="CUSTOM">
    <xs:annotation>
      <xs:documentation>Custom Attributes</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:attribute name="NAME" type="xs:string" use="optional"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="LODESTARUSAGE">
    <xs:annotation>
      <xs:documentation>root node</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="MDMUSAGE" type="stdMDMUsage"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="abstractMDMUsage" abstract="true">
    <xs:annotation>
      <xs:documentation>Abstract MDM Usage</xs:documentation>
    </xs:annotation>
    <xs:sequence maxOccurs="2">
      <xs:element ref="METERREADS" minOccurs="0"/>
      <xs:element ref="METEREVENTS" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="stdMDMUsage">
    <xs:annotation>
      <xs:documentation>Standard MDM Usage</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="abstractMDMUsage">
        <xs:attribute name="METERID" type="xs:string" use="required"/>
        <xs:attribute name="UOMCODE" type="xs:string" use="required"/>
        <xs:attribute name="CHANNELID" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="serialNumberedMDMUsage">
    <xs:annotation>
      <xs:documentation>Serial Number</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="abstractMDMUsage">
        <xs:attribute name="UOMCODE" type="xs:string" use="required"/>
        <xs:attribute name="HEADERSERIALNUMBER" type="xs:string"
use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

```

        <xs:attribute name="METERMANUFACTURER" type="xs:string" use="required"/
    >
        <xs:attribute name="EFFECTIVEDATE" type="xs:string" use="required"/>
        <xs:attribute name="CHANNELID" type="xs:string" use="optional"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="abstractRead" abstract="true">
    <xs:annotation>
        <xs:documentation>Abstract Read</xs:documentation>
    </xs:annotation>
    <xs:attribute name="TOTALVAL" type="xs:string" use="optional"/>
    <xs:attribute name="ENERGY" type="xs:string" use="optional"/>
    <xs:attribute name="DESCRIPTOR" type="xs:string" use="optional"/>
    <xs:attribute name="SPI" type="xs:string" use="optional"/>
    <xs:attribute name="METERMULTIPLIER" type="xs:string" use="optional"/>
    <xs:attribute name="METEROFFSET" type="xs:string" use="optional"/>
    <xs:attribute name="PULSEMULTIPLIER" type="xs:string" use="optional"/>
    <xs:attribute name="PULSEOFFSET" type="xs:string" use="optional"/>
    <xs:attribute name="POWERFACTOR" type="xs:string" use="optional"/>
    <xs:attribute name="DSTPARTICIPANT" type="xs:string" use="optional"/>
    <xs:attribute name="TOTALIZE" type="xs:string" use="optional"/>
    <xs:attribute name="DCFLOW" type="xs:string" use="optional"/>
    <xs:attribute name="ORIGIN" type="xs:string" use="optional"/>
    <xs:attribute name="WEIGHT" type="xs:string" use="optional"/>
    <xs:attribute name="POPULATION" type="xs:string" use="optional"/>
    <xs:attribute name="ACCEPTREJECTSTATUS" type="xs:string" use="optional"/>
    <xs:attribute name="TRANSLATIONTIME" type="xs:string" use="optional"/>
    <xs:attribute name="TZSTDNAME" type="xs:string" use="optional"/>
    <xs:attribute name="READDATE" type="xs:string" use="optional"/>
    <xs:attribute name="READMETHOD" type="xs:string" use="optional"/>
    <xs:attribute name="SERIALNUMBER" type="xs:string" use="optional"/>
    <xs:attribute name="TRANSFORMERLOSSFACTOR" type="xs:string" use="optional"/
    >
    <xs:attribute name="NUMOFDIALS" type="xs:string" use="optional"/>
    <xs:attribute name="DEFAULTDECIMALS" type="xs:string" use="optional"/>
</xs:complexType>
<xs:complexType name="consumptionRead">
    <xs:annotation>
        <xs:documentation>Consumption Read</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="abstractRead">
            <xs:attribute name="STOPREADTIME" type="xs:string" use="required"/>
            <xs:attribute name="STOPVAL" type="xs:string" use="required"/>
            <xs:attribute name="STARTREADTIME" type="xs:string" use="optional"/>
            <xs:attribute name="STARTVAL" type="xs:string" use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="intervalRead">
    <xs:annotation>
        <xs:documentation> Interval Read</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="abstractRead">
            <xs:sequence>
                <xs:element name="INTS" minOccurs="0">
                    <xs:complexType>
                        <xs:sequence maxOccurs="unbounded">
                            <xs:element name="I">
                                <xs:complexType>
                                    <xs:attribute name="V" type="xs:string" use="required"/>
                                    <xs:attribute name="S" type="xs:string" use="required"/>
                                    <xs:attribute name="E" type="xs:string" use="optional"/>
                                    <xs:attribute name="UD" type="xs:string" use="optional"/>
                                    <xs:attribute name="TS" type="xs:string" use="optional"/>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:sequence>
<xs:attribute name="STARTVAL" type="xs:string" use="optional"/>
<xs:attribute name="STOPVAL" type="xs:string" use="optional"/>
<xs:attribute name="STARTREADTIME" type="xs:string" use="required"/>
<xs:attribute name="STOPREADTIME" type="xs:string" use="optional"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="touRead">
  <xs:annotation>
    <xs:documentation>TOU Read</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="consumptionRead">
      <xs:sequence>
        <xs:element name="TOUPERIODS">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="TOUPERIOD" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                  <xs:attribute name="TOUPERIODNAME" type="xs:string"
use="required"/>
                  <xs:attribute name="STARTVAL" type="xs:double"
use="optional"/>
                  <xs:attribute name="STOPVAL" type="xs:double"
use="required"/>
                  <xs:attribute name="STARTREADTIME" type="xs:dateTime"
use="optional"/>
                  <xs:attribute name="STOPREADTIME" type="xs:dateTime"
use="required"/>
                  <xs:attribute name="TOTALVAL" type="xs:double"
use="optional"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="METERREADS">
  <xs:complexType>
    <xs:sequence maxOccurs="unbounded">
      <xs:element name="METERREAD" type="consumptionRead" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Consumption Read</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="METERREAD" type="intervalRead" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Interval Read</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="METERREAD" type="touRead" minOccurs="0">
        <xs:annotation>
          <xs:documentation>TOU Read</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="METEREVENTS">

```

```
<xs:complexType>
  <xs:sequence maxOccurs="unbounded">
    <xs:element name="METEREVENT">
      <xs:complexType>
        <xs:attribute name="STARTREADTIME" type="xs:string" use="required"/
>
        <xs:attribute name="STOPREADTIME" type="xs:string" use="required"/>
        <xs:attribute name="EVENTCODE" type="xs:string" use="required"/>
        <xs:attribute name="LOGDATE" type="xs:string" use="required"/>
        <xs:attribute name="NOTES" type="xs:string" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="MDMEVENTS">
  <xs:complexType>
    <xs:sequence maxOccurs="unbounded">
      <xs:element name="MDMEVENT">
        <xs:complexType>
          <xs:attribute name="SERIALNUMBER" type="xs:string" use="required"/>
          <xs:attribute name="MANUFACTURER" type="xs:string" use="optional"/>
          <xs:attribute name="EVENTCODE" type="xs:string" use="required"/>
          <xs:attribute name="LOGDATE" type="xs:string" use="required"/>
          <xs:attribute name="NOTES" type="xs:string" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

# Oracle Utilities Meter Data Management Business Rule Types

Oracle Utilities Meter Data Management uses the Energy Information Platform Adapter to perform various business functions, including import of usage data, performing transactions via the web user interface, and invoking of web services to communicate with external meter read systems. This section provides descriptions of the types of Adapter Business Rules used by Oracle Utilities Meter Data Management, including:

- **Usage Import Business Rules**
- **Web Transaction Business Rules**

## Usage Import Business Rules

Usage Import business rules are business rules configured to import usage data into Oracle Utilities Meter Data Management, and include:

- **Import/Usage**
- **MDMElsterImport**
- **MDMReady**

### Import/Usage

Import/Usage business rules are used to import usage readings into Oracle Utilities Meter Data Management from non-standard formats.

Business rules of this type use Rule Description Language to perform mapping from the external format into Oracle Utilities Meter Data Management. Specifically, Import/Usage business rules use RDLs of type “Import/Usage,” which include a pre-defined Template Destination comprising the following tables:

- **Meter Event:** Used to map meter events
- **Physical Meter Event:** Used to map physical meter events
- **Interval Cut:** Used to map header information for interval usage
- **Interval Read:** Used to map interval values, status codes, and (optionally) extended status codes from interval usage
- **Consumption Meter Reading:** Used to map consumption usage
- **Time of Use Meter:** Used to map consumption readings for time of use usage
- **Time of Use Period Reading:** Used to map time of use period readings for time of use usage

When configuring an RDL of this type, data from the payload format is mapped to the appropriate tables, in the same manner as when configuring RDLs of type Import or Import Interval. See **Configuring Rule Description Language** on page 11-62 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about configuring Rule Description Language.

**Note:** Import/Usage business rules require that the payload data be in (or converted into) an XML format.

## Import/Usage Business Rule Properties

Import/Usage business rules use the same properties as Import and Import Interval business rules. See **Import Properties and Parameters** on page 11-108 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about these properties.

In addition, Import/Usage business rules also use the following unique property.

### PROPERTY: calcreading

- **Required:** No
- **Possible Values:** "y" or "n" or "true" or "false"
- **Description:** This property specifies that RDL to calculate interval usage "on the fly" from a previous interval reading. The calculation for the first interval value is dependent upon the presence of a previous reading, and uses the STOPVAL of the previous reading for this calculation. This value is stored in a reserved location within the RDL called PTS\_STARTVAL. If no previous usage reading is available, then the first interval value is '0' and its status is set to '9'. The Echelon Load Profile RDL (ECHELON\_INTD\_RDL) used by the Echelon converter uses this property to either calculate each interval usage or not. The PTS\_STARTVAL can be populated with a single value from the result of a query or any of the other tools available in the RDL. However, the PTS\_STARTVAL must be part of the GLOBAL variable set in order for the first reading to be calculated. See **Echelon Converter** on page 5-25 for more information about the Echelon converter.

**Note:** This property should be used only with interval usage that needs to be calculated at run time.

- **Example:**

```
<PROPERTY NAME='calcreading' VALUE='y' />
```

### PROPERTY: uomchannelstocalc

- **Required:** No
- **Possible Values:** comma separated list of UOMs/channel IDs
- **Description:** This property is used in conjunction with the "calcreading" property to calculate interval usage "on the fly" for a specified list of UOMs and Channel ID combinations. This property behaves as follows:
  - If "calcreading" is enabled and "uomchannelstocalc" is undefined, values will be calculated for reads for any UOM.
  - If "calcreading" is enabled and "uomchannelstocalc" is defined, values will be calculated for reads with the UOMs defined in the property for all or specific CHANNELIDs. If there are no specific CHANNELID/s defined for a UOM, values will be calculated for the UOM regardless of CHANNELID.
  - If "calcreading" is disabled, whether "uomchannelstocalc" is defined or not, values will NOT be calculated at all.

The specific UOMs to apply calculation are defined in the property separated by commas (.). To apply calculation for specific Channel IDs for a UOM, immediately after the UOM enclose the desired Channel IDs within square brackets ([ ]) separated by vertical bars (|).

- **Examples:**

Calculate values for UOM codes 01, 02, and 03.

```
<PROPERTY NAME='uomchannelstocalc' VALUE='01,02,03' />
```

Calculate values for UOM code 01, code 02 for Channel IDs 0 or 1, and code 12 for Channel ID 1 only.

```
<PROPERTY NAME='uomchannelstocalc' VALUE='01,02[0|1],12[1]' />
```



## MDMElsterImport

MDMElsterImport business rules are used to convert and import Elster formatted usage data into Oracle Utilities Meter Data Management.

## Interval Usage

The table below illustrates the mapping and conversion from the Elster standard interval data file to the Oracle Utilities Meter Data (LSM) format.

Elster Format	Mapped to LSM Format
/AMRDEF/MeterReadings/Meter/ SerialNumber	/LODESTARUSAGE/MDMUSAGE/ SERIALNUMBER
/AMRDEF/MeterReadings/Meter/MeterIrn	/LODESTARUSAGE/MDMUSAGE/ MeterIrn
/AMRDEF/MeterReadings/Meter/ AccountIdent	/LODESTARUSAGE/MDMUSAGE/ AccountIdent
/AMRDEF/MeterReadings/Meter/ MeterName	/LODESTARUSAGE/MDMUSAGE/ METERNAME
/AMRDEF/MeterReadings/Meter/InstallDate	/LODESTARUSAGE/MDMUSAGE/ INSTALLDATE
/AMRDEF/MeterReadings/ ReadingQualityIndicator	/LODESTARUSAGE/METEREVENTS
/AMRDEF/MeterReadings/Meter/ IntervalData/IntervalSpec/Interval	/LODESTARUSAGE/MDMUSAGE/ METERREADS/METERREAD/SPI
/AMRDEF/MeterReadings/Meter/ IntervalData/IntervalSpec/UOM	/LODESTARUSAGE/MDMUSAGE/ METERREADS/METERREAD/ UOMCODE
/AMRDEF/MeterReadings/Meter/ IntervalData/IntervalSpec/Channel	/LODESTARUSAGE/MDMUSAGE/ METERREADS/METERREAD/ CHANNELID
/AMRDEF/MeterReadings/Meter/ IntervalData/IntervalSpec/TimeStampStart	/LODESTARUSAGE/MDMUSAGE/ METERREADS/METERREAD/ STARTREADTIME
/AMRDEF/MeterReadings/Meter/ IntervalData/IntervalSpec/TimeStampEnd	/LODESTARUSAGE/MDMUSAGE/ METERREADS/METERREAD/ STOPREADTIME
/AMRDEF/MeterReadings/Meter/ IntervalData/IntervalSpec/Multiplier	/LODESTARUSAGE/MDMUSAGE/ METERREADS/METERREAD/ METERMULTIPLIER
/AMRDEF/MeterReadings/Meter/ IntervalData/IntervalSpec/Reading/Value	/LODESTARUSAGE/MDMUSAGE/ METERREADS/METERREAD/INTS/I
/AMRDEF/MeterReadings/ OutageCountSummary	/LODESTARUSAGE/METEREVENTS

**Consumption/Time of Use Usage**

The table below illustrates the mapping and conversion from the Elster standard consumption/TOU usage file to the Oracle Utilities Meter Data (LSM) format.

<b>Elster Format</b>	<b>Mapped to LSM Format</b>
/AMRDEF/MeterReadings/Meter/ SerialNumber	/LODESTARUSAGE/MDMUSAGE/ SERIALNUMBER
/AMRDEF/MeterReadings/Meter/MeterIrn	/LODESTARUSAGE/MDMUSAGE/ MeterIrn
/AMRDEF/MeterReadings/Meter/ AccountIdent	/LODESTARUSAGE/MDMUSAGE/ AccountIdent
/AMRDEF/MeterReadings/Meter/ MeterName	/LODESTARUSAGE/MDMUSAGE/ METERNAME
/AMRDEF/MeterReadings/Meter/InstallDate	/LODESTARUSAGE/MDMUSAGE/ INSTALLDATE
/AMRDEF/MeterReadings/ ReadingQualityIndicator	/LODESTARUSAGE/METEREVENTS
/AMRDEF/MeterReadings/Meter/ ConsumptionData/ConsumptionSpec/UOM	/LODESTARUSAGE/MDMUSAGE/ METERREADS/METERREAD/UOM
/AMRDEF/MeterReadings/Meter/ ConsumptionData/ConsumptionSpec/ Multiplier	/LODESTARUSAGE/MDMUSAGE/ METERREADS/METERREAD/ METERMULTIPLIER
/AMRDEF/MeterReadings/Meter/ ConsumptionData/Reading/Value	/LODESTARUSAGE/MDMUSAGE/ METERREADS/METERREAD/ STOPVAL
/AMRDEF/MeterReadings/Meter/ ConsumptionData/Reading/RawReading	/LODESTARUSAGE/MDMUSAGE/ METERREADS/METERREAD/ STOPVAL
/AMRDEF/MeterReadings/Meter/ ConsumptionData/Reading/TimeStamp	/LODESTARUSAGE/MDMUSAGE/ METERREADS/METERREAD/ STOPREADTIME
/AMRDEF/MeterReadings/Meter/ ConsumptionData/Reading/QualityFlags	/LODESTARUSAGE/METEREVENTS
/AMRDEF/MeterReadings/Meter/ ConsumptionData/EventData	/LODESTARUSAGE/METEREVENTS

An example Business Rule of this type called “BR\_ELSTER\_IMPORT” is provided in the Oracle Utilities Data Repository.

**MDMElsterImport and Power Flow Direction**

Metered energy can flow in one of two directions: from the grid to the customer, or from the customer to the grid (in cases where the customer is able to generate some of their own electricity). In Oracle Utilities Meter Data Management (and in the Energy Information Platform and other related Oracle Utilities products), this direction is referred to as Power Flow Direction, and is defined in terms relative to the grid. That is, energy flowing from the grid to the customer is referred to as Distributed ("D"), while energy flowing from the customer to the grid is referred to as Received ("R"). The direction which meter energy flows for a given meter is defined by the

Power Flow Direction attribute on the Meter Configuration record. This can also be defined in individual usage readings, and in the LSM format is defined by the DCFLOW attribute.

In Elster formatted data, the Power Flow Direction is not limited to Received or Distributed, and can be defined as any string of characters. Because of this, the application cannot always properly map the direction defined in Elster readings to the correct Oracle Utilities Meter Data Management direction.

Because interval data is always channel-specific, it's possible to use channel numbers to specify Power Flow Direction. For example, channel 1 could be used to represent Delivered energy (flowing from the grid to the customer), while channel 2 could be used to represent Received energy (flowing from the customer to the grid).

Consumption and Time of Use data is not typically channel-specific, it's necessary to define mappings between the Direction specified in the Elster file and the channel ID and Oracle Utilities Meter Data Management Power Flow Direction

These mappings can be defined using the following properties:

- **TO\_GRID**: designates energy flowing from the client to the grid
- **FROM\_GRID**: designates energy flowing from the grid to the client.

The format for specifying these properties is as follows:

```
VALUE = "Direction=[DIRECTION];Channel=[CHAN_ID];DCFlow=[DCFLOW]"
```

where:

- DIRECTION is the value that will be used in the Elster file for the 'Direction' attribute.
- CHAN\_ID is the Channel Id of the meter where the data for the given direction should be stored.
- DCFLOW is the Power Flow Direction (DCFLOW) value to use for readings with the specified DIRECTION.

Example:

```
<PROPERTY NAME="TO_GRID" VALUE="Direction=Delivered;Channel=01;DCFlow=R" />
<PROPERTY NAME="FROM_GRID" VALUE="Direction=Received;Channel=02;DCFlow=D" />
```

In the example above, the TO\_GRID property specifies that within the Elster file, for all ConsumptionData elements that contain a ConsumptionSpec element with the 'Direction' attribute with a value of "Delivered" should be mapped to a logical meter with ChannelId='01', and set the Power Flow Direction flag to "R".

- 1) If neither property is specified, the business rule will assume that the data is uni-directional, and will not use Channel Id at all. The Power Flow Direction will be set to 'R' (the default value).
- 2) If either property is specified in the properties and a ConsumptionSpec element with the 'Direction' attribute is found in an Elster file with a value not specified in the properties, an exception will be triggered.

## MDMReady

MDMReady business rules are used to import usage readings into Oracle Utilities Meter Data Management from the Oracle Utilities Meter Data (LSM) Format. See **Oracle Utilities Meter Data (.LSM) File Format** on page 5-2 for a detailed description of this format.

Consumption and Time of Use (TOU) usage must be in (or converted into) the Oracle Utilities Meter Data (LSM) format when using MDMReady business rules.

Interval usage can be provided in (or converted into) the Oracle Utilities Meter Data (LSM) format, or in one of the standard Oracle Utilities interval data formats. In the latter case, the Interval Data (MV90) Converter is required. See **MV90 Interval Data Converter** on page 5-17

for more information about the Interval Data Converter. See **Using Adapter Data Converters** on page 6-23 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about using Oracle Utilities Adapter data converters.

### **MDMReady Business Rule Properties**

MDMReady business rules use the same properties as Import and Import Interval business rules. See **Import Properties and Parameters** on page 11-108 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about these properties.

## **Web Transaction Business Rules**

Web transaction business rules execute specific types of operations performed by users via the web-based user interface. These operations include:

- Importing/uploading usage data (via LSM files) using the **Upload Files** screen
- Saving changes made to usage readings using the **Edit Usage**, **Edit TOU Usage**, or **Edit Values** screens.
- Manually validating and/or approving usage readings using the **Edit Usage**, **Edit TOU Usage**, or **Edit Values** screens.

Web transaction business rules are MDMReady business rules, and **MUST** be executed by Runtime Services of type MDMPORTAL.

**Note:** At least one runtime service of type MDMPORTAL must be running in order to perform any of the above operations.

# Oracle Utilities Meter Data Management Data Converters

Oracle Utilities Meter Data Management provides several “out of the box” converters that can be used to convert usage data produced by meter vendors into a format usable by Oracle Utilities Meter Data Management. These converters include:

- **MV90 Interval Data Converter**
- **MV9 Interval Data Converter**
- **LSE Interval Data Converter**
- **LegacyMV90 Interval Data Converter**
- **Selecting the Appropriate Meter Data Management Interval Data Converter**
- **Echelon Converter**
- **CellNet Converter**
- **MVRS Converter**

## MV90 Interval Data Converter

The MV90 Interval Data Converter is a Java class that converts interval data into the Oracle Utilities Meter Data (LSM) format used by the Oracle Utilities Meter Data Management application. The `com.iodestarcorp.portal.data` Java package contains this class (MV90Converter).

The MV90 Interval Data Converter converts interval data in any of the Oracle Utilities supported interval data formats, including:

- Oracle Utilities Enhanced Input/Output Interval Data Format (\*.lse)
- Oracle Utilities Standard Interval Data Format (\*.inp)
- Oracle Utilities Comma Separated Interval Data Format (\*.csv)
- Oracle Utilities Standard XML Interval Data Format (\*.xml)
- MV90 Mainframe Format (\*.mv9)
- EU Engineering Units Format (\*.eu)
- DTV Engineering Units Format (\*.dtv)
- Meter Data Exchange Format (\*.mdf)

See **Appendix C: Oracle Utilities Enhanced Input/Output Interval Data Format**, **Appendix D: Oracle Utilities Standard XML Interval Data Format**, and **Appendix E: Oracle Utilities Comma Separated Values (CSV) Interval Data Format** in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about these interval data formats.

**Arguments:** The MV90Converter accepts the following arguments

Argument	Description
operatingExtension	The file extension for the expected type of payload. Options could include mv9, lse, and xml. The default is mv9.

Argument	Description
meterIdAvailable	This argument determines what data is in the RECORDER field of the incoming data. If the RECORDER is the MDM MeterId, the meterIdAvailable argument should be set to true. If set to false, the data converter assumes that the RECORDER contains the serial number for the meter. The default value is false.
debugging	This true or false parameter enables or disables debugging mode. In normal operation this parameter would be set this to false, but when true the temporary files are not deleted. The default value is false.
configurationPath	When provided, this argument is used to specify the location of the lodestar.cfg file which can be used to configure certain properties of intdexp.exe. By default, this argument is unused.

## Using the MV90 Converter with Adapter Services

To use the MV90 Interval Data Converter with an Adapter service, you must specify the converter in Service Properties:

- Input Data Converter (**INPUT\_DC\_n**)

See **Defining Service Properties for Adapter Runtime Services** on page 11-28 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about service properties.

## Format

The format for specifying converters is:

```
<PROPERTY NAME="INPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.MV90Converter[,<FILE_EXTENSION>,
<ARGUMENTS>]" />
```

where:

- **<FILE\_EXTENSION>** is the name of an optional file format extension (lse, mdf, etc.) for the data to be converted. The default file format is MV90 (mv9), and need not be specified when converting data from MV90 format.
  - When specifying a file format, you must include the comma between the classname (MV90Converter) and the file format extension. If not specifying a file format, the comma should NOT be included.
- **<ARGUMENTS>** are optional arguments used by the converter. Arguments are optional, but arguments cannot be skipped. For example, if a data converter has four arguments, the first two must be supplied in order to specify the third.

### Examples:

Use the MV90 Interval Data Converter to convert an MV90 file.

```
<PROPERTY NAME="INPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.MV90Converter" />
```

Use the MV90 Interval Data Converter to convert an LSE file.

```
<PROPERTY NAME="INPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.MV90Converter,lse"/>
```

### INTDCONFIG.CFG.XML Configuration File

When using this converter, the `<MV90StatusMapSource>` element in the INTDCONFIG.CFG.XML configuration file must be defined. This element specifies the data source where the MV90 Status Code Map table used to map MV-90 status codes to LODESTAR status codes on import of MV90 data is located. See **MV90 Status Code Map** on page 3-15 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about the data stored in this table.

## MV9 Interval Data Converter

The MV9 Interval Data Converter is a Java class that converts interval data in the MV90 format into the Oracle Utilities Meter Data (LSM) format used by the Oracle Utilities Meter Data Management application. The `com.lodestarcorp.portal.data.intd` Java package contains this class (MV9Converter).

**Arguments:** The MV9Converter accepts the following arguments

Argument	Description
recorderIsSerialNumber	This argument determines what data is in the RECORDER field of the incoming data. If the RECORDER is the MDM Serial Number, this argument should be set to true. If set to false, the data converter assumes that the RECORDER contains the (logical) Meter ID for the meter. The default value is true.

### Using the MV9 Converter with Adapter Services

To use the MV9 Interval Data Converter with an Adapter service, you must specify the converter in Service Properties:

- Input Data Converter (`INPUT_DC_n`)

See **Defining Service Properties for Adapter Runtime Services** on page 11-28 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about service properties.

### Format

The format for specifying converters is:

```
<PROPERTY NAME="INPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.intd.MV9Converter[ ,<ARGUMENTS> ]"/>
```

where:

- **<ARGUMENTS>** are optional arguments used by the converter. Arguments are optional, but arguments cannot be skipped. For example, if a data converter has four arguments, the first two must be supplied in order to specify the third.

#### Examples:

Use the MV9 Interval Data Converter to convert an MV90 file in which the Recorder ID is the MDM serial number.

```
<PROPERTY NAME="INPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.intd.MV9Converter"/>
```

Use the MV9 Interval Data Converter to convert an MV90 file in which the Recorder ID is NOT the MDM serial number.

```
<PROPERTY NAME="INPUT_DC_0"  
VALUE="com.lodestarcorp.portal.data.intd.MV9Converter,false"/>
```



## LSE Interval Data Converter

The LSE Interval Data Converter is a Java class that converts interval data in the Oracle Utilities Enhanced Input/Output Interval Data Format (\*.lse) format into the Oracle Utilities Meter Data (LSM) format used by the Oracle Utilities Meter Data Management application. The **com.lodestarcorp.portal.data.intd** Java package contains this class (LSEConverter).

See **Appendix C: Oracle Utilities Enhanced Input/Output Interval Data Format** in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about this interval data format.

**Arguments:** The LSEConverter accepts the following arguments

Argument	Description
recorderIsSerialNumber	This argument determines what data is in the RECORDER field of the incoming data. If the RECORDER is the MDM Serial Number, this argument should be set to true. If set to false, the data converter assumes that the RECORDER contains the (logical) Meter ID for the meter. The default value is true.

### Using the LSE Converter with Adapter Services

To use the LSE Interval Data Converter with an Adapter service, you must specify the converter in Service Properties:

- Input Data Converter (**INPUT\_DC\_n**)

See **Defining Service Properties for Adapter Runtime Services** on page 11-28 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about service properties.

### Format

The format for specifying converters is:

```
<PROPERTY NAME="INPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.intd.LSEConverter[ ,<ARGUMENTS> ]" />
```

where:

- **<ARGUMENTS>** are optional arguments used by the converter. Arguments are optional, but arguments cannot be skipped. For example, if a data converter has four arguments, the first two must be supplied in order to specify the third.

### Examples:

Use the LSE Interval Data Converter to convert an LSE file in which the Recorder ID is the MDM serial number.

```
<PROPERTY NAME="INPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.intd.LSEConverter" />
```

Use the LSE Interval Data Converter to convert an LSE file in which the Recorder ID is NOT the MDM serial number.

```
<PROPERTY NAME="INPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.intd.LSEConverter,false" />
```

## LegacyMV90 Interval Data Converter

The LegacyMV90 Interval Data Converter is a Java class that converts interval data into the Oracle Utilities Meter Data (LSM) format used by the Oracle Utilities Meter Data Management application. The `com.lodestarcorp.portal.data` Java package contains this class (LegacyMV90Converter).

The LegacyMV90 Interval Data Converter converts interval data in any of the Oracle Utilities supported interval data formats, including:

- Oracle Utilities Enhanced Input/Output Interval Data Format (\*.lse)
- Oracle Utilities Standard Interval Data Format (\*.inp)
- Oracle Utilities Comma Separated Interval Data Format (\*.csv)
- Oracle Utilities Standard XML Interval Data Format (\*.xml)
- MV90 Mainframe Format (\*.mv9)
- EU Engineering Units Format (\*.eu)
- DTV Engineering Units Format (\*.dtv)
- Meter Data Exchange Format (\*.mdf)

See **Appendix C: Oracle Utilities Enhanced Input/Output Interval Data Format**, **Appendix D: Oracle Utilities Standard XML Interval Data Format**, and **Appendix E: Oracle Utilities Comma Separated Values (CSV) Interval Data Format** in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about these interval data formats.

**Arguments:** The LegacyMV90Converter accepts the following arguments

Argument	Description
operatingExtension	The file extension for the expected type of payload. Options could include mv9, lse, and xml. The default is mv9.
meterIdAvailable	This argument determines what data is in the RECORDER field of the incoming data. If the RECORDER is the MDM MeterId, the meterIdAvailable argument should be set to true. If set to false, the data converter assumes that the RECORDER contains the serial number for the meter. The default value is false.
debugging	This true or false parameter enables or disables debugging mode. In normal operation this parameter would be set to false, but when true the temporary files are not deleted. The default value is false.
configurationPath	When provided, this argument is used to specify the location of the lodestar.cfg file which can be used to configure certain properties of intdexp.exe. By default, this argument is unused.

### Using the LegacyMV90 Converter with Adapter Services

To use the LegacyMV90 Interval Data Converter with an Adapter service, you must specify the converter in Service Properties:

- Input Data Converter (**INPUT\_DC\_n**)

See **Defining Service Properties for Adapter Runtime Services** on page 11-28 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about service properties.

## Format

The format for specifying converters is:

```
<PROPERTY NAME="INPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.LegacyMV90Converter[ ,<FILE_EXTENSION> ,
<ARGUMENTS>]" />
```

where:

- **<FILE\_EXTENSION>** is the name of an optional file format extension (lse, mdx, etc.) for the data to be converted. The default file format is MV90 (mv9), and need not be specified when converting data from MV90 format.
  - When specifying a file format, you must include the comma between the classname (MV90Converter) and the file format extension. If not specifying a file format, the comma should NOT be included.
- **<ARGUMENTS>** are optional arguments used by the parser. Arguments are optional, but arguments cannot be skipped. For example, if a data converter has four arguments, the first two must be supplied in order to specify the third.

### Examples:

Use the LegacyMV90 Interval Data Converter to convert an MV90 file.

```
<PROPERTY NAME="INPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.LegacyMV90Converter" />
```

Use the LegacyMV90 Interval Data Converter to convert an LSE file.

```
<PROPERTY NAME="INPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.LegacyMV90Converter,lse" />
```

## Selecting the Appropriate Meter Data Management Interval Data Converter

Oracle Utilities Meter Data Management offers several converters that can be used to convert interval data into the Oracle Utilities Meter Data (LSM) format. This section offers guidelines for selecting the most appropriate interval data converter for your implementation of Oracle Utilities Meter Data Management.

### MV90 Interval Data Converter

The MV90 Interval Data Converter is designed to provide improved performance over the MV90 Converter supported in previous versions of Oracle Utilities Meter Data Management. This converter supports importing usage data in any of the supported interval data formats (see **MV90 Interval Data Converter** on page 5-17).

This converter is appropriate when upgrading an existing Oracle Utilities Meter Data Management implementation (from v4.53, v4.54, or v1.5). Using this converter requires no change in configuration when upgrading from previous versions of Oracle Utilities Meter Data Management.

## MV9 Interval Data Converter

The MV9 Interval Data Converter is designed to provide improved performance over the MV90 Converter supported in previous versions of Oracle Utilities Meter Data Management when importing usage data in the MV90 Mainframe format.

This converter is appropriate for new implementations of Oracle Utilities Meter Data Management when usage data to be imported is in the MV90 Mainframe Format format.

## LSE Interval Data Converter

The LSE Interval Data Converter is designed to provide improved performance over the MV90 Converter supported in previous versions of Oracle Utilities Meter Data Management when importing usage data in the Oracle Utilities Enhanced Input/Output Interval Data Format (\*.lse) format.

This converter is appropriate for new implementations of Oracle Utilities Meter Data Management when usage data to be imported is in the Oracle Utilities Enhanced Input/Output Interval Data Format (\*.lse) format.

## LegacyMV90 Interval Data Converter

The LegacyMV90 Interval Data Converter is designed to provide the same performance as the MV90 Converter supported in previous versions of Oracle Utilities Meter Data Management when importing usage data in any of the any of the supported interval data formats (see **MV90 Interval Data Converter** on page 5-17).

Use of this converter is only appropriate if use of one of the above converters is not possible due to some issue related to your implementation and/or configuration of Oracle Utilities Meter Data Management.

## Summary

The table below summarizes the above guidelines for use of the interval data converters provided with Oracle Utilities Meter Data Management.

For this situation...	Use this Interval Data Converter
Upgrading from v4.53 or v1.5 to v1.6	<b>MV90 Interval Data Converter</b> (requires no change in configuration)
New Implementation, importing data in MV90 format	<b>MV9 Interval Data Converter</b>
New Implementation, importing data in LSE format	<b>LSE Interval Data Converter</b>
New implementation, importing data in other supported format	<b>MV90 Interval Data Converter</b>

## Echelon Converter

The Echelon converter is a set of pre-configured RDLs and Business Rules used to import Echelon usage readings into Oracle Utilities Meter Data Management.

Echelon readings are received as 2 files, one containing register readings (also referred to as the billing file) and interval usage (also referred to as the load profile file).

- **Billing File:** The billing file is composed of 4 tiers, each representing a different tariff or time-of-use period such as On Peak or Off Peak. The billing file also includes a summary tier that is the total of all the other tiers in the file.
- **Load Profile File:** The load profile file include interval data for the meter in watt hours, and can include up to 8 channels of data. Data is received in UTC time, but is converted into the local time zone of the meter. The file is also checked for missing intervals. Missing intervals are identified with a value of 0 (zero) and a status code of 9.

There are three primary components of the Echelon converter:

- **XSLT Converters**
- **Rule Description Language**
- **Billing Rules**

### XSLT Converters

The Echelon converter makes use to two XSLT converters, installed with the application, that are used to covert the incoming Echelon formatted data into a format more usable with Rule Description Language configuration.

#### Billing File

The XSLT converter used with the billing file transforms the Echelon billing payload into an XML payload usable by the pre-configured Rule Description Language (see below). This converter also creates meter events for power outages, and a meter event when the error count within the original payload is greater than zero.

**Location:** C:\Lodestar\Bin\billing.xslt

#### Load Profile File

The XSLT converter used with the load profile file transforms the Echelon load profile payload into an XML payload usable by the pre-configured Rule Description Language (see below).

**Location:** C:\Lodestar\Bin\load\_profile.xslt

### Rule Description Language

The Echelon converter makes use to two Rule Description Language configurations, installed with the application, that are used to map the incoming Echelon formatted data into the Oracle Utilities Data Repository and the Oracle Utilities Meter Data Management application.

#### Billing File

The RDL used with the billing file is called ECHELON\_BILLING\_RDL, and is defined as follows:

Rule Description Language:

- **RDL Name:** ECHELON\_BILLING\_RDL
- **Rule Type Code:** Import/Usage

RDL Version:

- **RDL Name:** ECHELON\_BILLING\_RDL
- **Start Time:** 09/09/2008 12:57:00

- **Stop Time:** N/A
- **Note:**

### Load Profile File

The RDL used with the load profile file is called ECHELON\_INTD\_RDL, and is defined as follows:

Rule Description Language:

- **RDL Name:** ECHELON\_INTD\_RDL
- **Rule Type Code:** Import/Usage

RDL Version:

- **RDL Name:** ECHELON\_INTD\_RDL
- **Start Time:** 09/09/2008 12:57:00
- **Stop Time:** N/A
- **Note:**

### Billing Rules

The Echelon converter also makes use to two Business Rules, installed with the application, that are used to define the import of Echelon formatted data into the Oracle Utilities Data Repository and the Oracle Utilities Meter Data Management application.

### Billing File

The Business Rule used with the billing file is called ECHELON\_BILLING\_RULE, and is defined as follows:

Business Rule:

- **Rule Name:** ECHELON\_BILLING\_RULE
- **Description:** Echelon Billing Rule
- **Rule Type:** Import/Usage
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME='xmlstream' VALUE='n' />
  <PROPERTY NAME='mode' VALUE='event' />
  <PROPERTY NAME='XslConverter' VALUE='c:\lodestar\bin\billing.xslt' />
</PROPERTIES>
```

Note that the above properties reference the billing XSLT converter described above.

Business Rule to RDL Association:

- **Business Rule:** ECHELON\_BILLING\_RULE
- **RDL Name:** ECHELON\_BILLING\_RDL

### Load Profile File

The Business Rule used with the load profile file is called ECHELON\_INTD\_RULE, and is defined as follows:

Business Rule:

- **Rule Name:** ECHELON\_INTD\_RULE
- **Description:** Echelon Load Profile Rule
- **Rule Type:** Import/Usage

- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME='xmlstream' VALUE='n' />
  <PROPERTY NAME='mode' VALUE='event' />
  <PROPERTY NAME='calcreading' VALUE='y' />
  <PROPERTY NAME='XslConverter' VALUE='c:\lodestar\bin\load_profile.xslt' />
</PROPERTIES>
```

Note that the above properties reference the load profile XSLT converter described above.

Business Rule to RDL Association:

- **Business Rule:** ECHELON\_INTD\_RULE
- **RDL Name:** ECHELON\_INTD\_RDL

## Using the Echelon Converter with Adapter Services

To use the Echelon converter, configure Adapter Runtime Services for both the Billing and Load Profile payloads.

## A Note About Configuring Echelon Meters

Echelon meters must be configured as meters with a Usage Type Code of Interval when using this configuration. In addition, because of the manner in which Echelon readings are received (billing file and load profile file), the use of this configuration requires that the Register Reading validation be configured for all Echelon meters.

## CellNet Converter

The CellNet converter is a Java class that converts consumption data from the CellNet format into the Oracle Utilities Meter Data (LSM) format used by the Oracle Utilities Meter Data Management application. The **com.lodestarcorp.custom** Java package contains this class (CnetConverter).

The table below outlines how data from the CellNet format is mapped into the LSM format. In this table:

- **Field:** The name of the field in the CellNet file
- **Description:** A description of the field
- **Start:** The starting position of the data in the CellNet file
- **LSM Element:** The element or attribute in the LSM file to which the field is mapped.
- **Example:** An example of the field data
- **Notes:** Notes specific to the mapping of the field.

Field	Description	Start	LSM Element	Example	Notes
METER ID	Meter ID	1	METERID	1224827406621	
UTILITY ID	Utility Id	21	NOT USED	Do not map	
UNITS	UOM	41	UOM CODE	KWH	
NOT IN FILE	Not in incoming file	-	USAGETYPECODE = CONSUMPTION	Default the code to 'CONSUMPTION' for all cellnet files.	
READ DESCRIPTION	Type of Read CUM, TOU	47	NOT USED	Do not map	

Field	Description	Start	LSM Element	Example	Notes
READ TYPE	B-Billing, D-Daily	57	NOT USED	Do not map	
NOT IN FILE	Not in incoming file	-	NEW-METERREADSYSTEMLOADTIME	31-JAN-06	To be calculated based on the date when the file import occurs.
READ TIME	Read Time	58	STOPREADTIME	30-JAN-06 23:59:59	
NOT IN FILE	Not in incoming file	-	STARTREADTIME	30-JAN-06 00:00:00	To be calculated as the previous READTIME plus 1 second in the system for that Meter ID and UOM.
READ VALUE	Consumption KWH	72	STOPVALUE	9999999.999	
NOT IN FILE	Not in incoming file	-	STARTVALUE	12345678.901	To be calculated as the previous STOPVALUE in the system for that Meter ID and UOM.
NOT IN FILE	Not in incoming file	-	TOTALVALUE, ENERGY	87654321.098	To be calculated as the STOPVALUE minus the STARTVALUE.
MAGNETIC FLAG	Magnetic Flag (Y/N)	84	NEW-MAGNETICFLAG	Y	
REVERSE ROTATION FLAG	Reverse Rotation (Y/N)	85	NEW-REVERSEROTFLAG	Y	
POWER OUT FLAG	Power Out Flag (Y/N)	86	NEW-POWEROUTFLAG	Y	
COVER OFF FLAG	Cover Removed (Y/N)	87	NEW-COVEROFFFLAG	Y	
POWER OUTAGE COUNT	Number of Power Outages in a day	88	NEW--NUMBERPOWEROUTAGES	99999	
READ METHOD	Read Type (Auto or Manual)	93	READMETHOD	A	
LOGICAL DISCONNECT	Indicates whether meter exceeds the daily Cumulative read total. (Y/N)	94	NEW--LOGICALDISCONNECT	Y	



## Using the CellNet Converter with Adapter Services

To use the CellNet converter with an Adapter service, you must specify the converter in Service Properties:

- Input Data Converter (**INPUT\_DC\_n**)

See **Defining Service Properties for Adapter Runtime Services** on page 11-28 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about service properties.

### Format

The format for specifying converters is:

```
<PROPERTY NAME="INPUT_DC_0" VALUE="com.lodestarcorp.custom.CnetConverter" />
```

#### Example:

Use the CellNet Converter to convert an CellNet file.

```
<PROPERTY NAME="INPUT_DC_0" VALUE="com.lodestarcorp.custom.CnetConverter" />
```

## MVRS Converter

The MVRS converter is a Java class that converts usage data from the MVRS format into the Oracle Utilities Meter Data (LSM) format used by the Oracle Utilities Meter Data Management application. The **com.lodestarcorp.custom.data** Java package contains this class (MvrsConverter).

### Mapping from MVRS to LSM

Below is an example of what a MVRS document looks like.

```
FHDNNNN020001389001491001728N
004751
```

```
RHD202007XXNN0000000300010000000000000010002000000NY0001000200030000000000
000000000200000000JA 20070427NN000000 N
```

```
CUS202081XX00172026218512MGSHNONR xxxx,xxxxxx 1111 N whatever st
001N 01
```

```
MTR202081XX002001000000000000CA 01635971
000100000220000002200030000000NN NNE 000 47421 1C 5122840781
```

```
RDG202081XXKW
```

```
1R000000000K00004000000000058011504272007132528N00000000000N6355 001080502K
00000400010000000000
```

```
RDG202081XXKWH2R000000000K00004000000011918011604272007132531N000000000000
N6355 000030500K 00000400020000011702
```

```
RTR20410WXXNN0000000700000000007000000010001000000NY0001000100070000000000
000000000100000000JA 20070427NN000000 N
```

```
FTRNNNN020001389001491001728N
```

The MVRS converter ignores any line that does not begin with the first three(3) characters 'MTR' or 'RDG'.

### Meter (MTR) Records

The MVRS converter ignores any line that does not begin with the first three(3) characters 'MTR' or 'RDG'. In the example file above, the first MTR record is as follows:

```
MTR202081XX002001000000000000CA 01635971
000100000220000002200030000000NN NNE 000 47421 1C 5122840781
```

In each MTR line:

- The Meter ID (METERID) is defined in columns 46 through 57.

```
MTR202081XX002001000000000000CA          01635971
000100000220000002200030000000NN  NNE 000  47421  1C 5122840781
```

- Additional information (AddInfo) is located in columns 102 through 106.

```
MTR202081XX002001000000000000CA          01635971
000100000220000002200030000000NN  NNE 000  47421  1C 5122840781
```

In the output LSM file, AddInfo and METERID are concatenated to form the HEADERSERIALNUMBER attribute.

4742101635971

## Reading (RDG) Records

In the example file above, the first RDG record is as follows:

```
RDG202081XXKW
1R000000000K00004000000011918011504272007132528N00000000000N6355  001080502K
00000400010000000000
```

In each RDG line:

- The UOM is defined in columns 12 through 15.

```
RDG202081XXKW
1R000000000K00004000000011918011504272007132528N00000000000N6355
001080502K 00000400010000000000
```

- Channel ID is located in columns 23 through 25.

```
RDG202081XXKW
1R000000001K00004000000011918011504272007132528N00000000000N6355
001080502K 00000400010000000000
```

- Read Method is located in column 26.

```
RDG202081XXKW
1R000000000K00004000000011918011504272007132528N00000000000N6355
001080502K 00000400010000000000
```

- Meter Multiplier is located in columns 27 through 32.

```
RDG202081XXKW
1R000000000K0004000000011918011504272007132528N00000000000N6355
001080502K 00000400010000000000
```

- Meter Reading is located in columns 34 through 43.

```
RDG202081XXKW
1R000000000K0000400000011918011504272007132528N00000000000N6355
001080502K 00000400010000000000
```

- Read Date is located in columns 48 through 55.

```
RDG202081XXKW
1R000000000K00004000000011918011504272007132528N00000000000N6355
001080502K 00000400010000000000
```

- Read Time is located in columns 56 through 61.

```
RDG202081XXKW
1R000000000K00004000000011918011504272007132528N00000000000N6355
001080502K 00000400010000000000
```

- Trouble Code 1 is located in columns 68 through 69.

```
RDG202081XXKW
1R000000000K00004000000011918011504272007132528N000000000000N6355
001080502K 00000400010000000000
```

- Trouble Code 2 is located in columns 70 through 72.

```
RDG202081XXXK
1R000000000K00004000000011918011504272007132528N000000000000N6355
001080502K 00000400010000000000
```

- New Read Method is located in column 94.

```
RDG202081XXXK
1R000000000K00004000000011918011504272007132528N000000000000N6355
001080502K 00000400010000000000
```

- Meter Constant is located in columns 97 through 102.

```
RDG202081XXXK
1R000000000K00004000000011918011504272007132528N000000000000N6355
001080502K 00000400010000000000
```

For each RDG record, there will be a separate MDMUSAGE node with meter read data included in the converted LSM file.

Based on the above sample data, the MVRS Converter would create an LSM file that would look like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<MDMUSAGE CHANNELID="1" EFFECTIVEDATE="2007-04-27T13:25:28"
HEADERSERIALNUMBER="4742101635971" METERID=" " UOMCODE="01"
USAGETYPE="CONSUMPTION">
  <METERREADS>
    <METERREAD ENERGY=" " METERMULTIPLIER="40" READMETHOD="K" STARTREADTIME=" "
STARTVAL=" " STOPREADTIME="2007-04-27T13:25:28" STOPVAL="11918" TOTALVAL=" "
TROUBLECODE1="00" TROUBLECODE2="00"/>
  </METERREADS>
</MDMUSAGE>
```

The MDMUSAGE attributes in the above example are as follows:

- CHANNELID directly comes from the Channel ID mapping described above (leading zeros are removed).
- EFFECTIVEDATE comes from the ReadDate and RateTime values mapping described above.
- HEADERSERIALNUMBER is the value described as the meter id (concatenation of AddInfo and METERID).
- METERID is always left blank.
- UOMCODE is the UOM mapping translated into the UOMCODE. This mapped value does not **come** from the database, but is translated by the converter.
- USAGETYPE is always CONSUMPTION.

The METERREAD attributes in the above example are as follows:

- ENERGY is always blank.
- METERMULTIPLIER comes from the MtrConstant mapping described above.
- READMETHOD comes from the NewReadMeth mapping described above.
- STARTREADTIME is always blank.
- STARTVAL is always blank.
- STOPREADTIME is the ReadDate and the ReadTime mapping described above.
- STOPVAL is always blank when the UOMCODE = '02', otherwise, it is the value of the MeterReading mapping described above.

- TOTALVAL is the MeterReading mapping described above if the UOMCODE - '02', otherwise, TOTALVAL is always blank.
- TROUBLECODE1 is the TroubleCode1 mapping described above.
- TROUBLECODE2 is the TroubleCode2 mapping described above.

### Using the MVRS Converter with Adapter Services

To use the MVRS converter with an Adapter service, you must specify the converter in Service Properties:

- Input Data Converter (**INPUT\_DC\_n**)

See **Defining Service Properties for Adapter Runtime Services** on page 11-28 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about service properties.

### Format

The format for specifying converters is:

```
<PROPERTY NAME="INPUT_DC_0"  
VALUE="com.lodestarcorp.custom.data.MvrsConverter" />
```

#### Example:

Use the MVRS Converter to convert an Mvrs file.

```
<PROPERTY NAME="INPUT_DC_0"  
VALUE="com.lodestarcorp.custom.data.MvrsConverter" />
```

## Configuring Usage Data Import using the Adapter

The Oracle Utilities Meter Data Management application uses the Energy Information Platform Adapter for automated data import. It also uses the Adapter when users perform upload, save, validate, and approve actions on usage data. via the web-based user interface This section describes how to set up the Adapter to process usage data for Oracle Utilities Meter Data Management, including:

- **Implementing Import/Usage Services and Business Rules**
  - **Import/Usage Business Rule Example**
- **Implementing MDMElsterImport Services and Business Rules**
  - **MDMElsterImport Business Rule Example**
- **Implementing MDMReady Services and Business Rules**
  - **MDMReady Business Rule Example - Usage Import**
  - **MDMReady Business Rule Example - Web Transactions**
- **Configuring Partial Commit and Error Management of Import Files**
- **Oracle Utilities Meter Data Management Adapter Properties**

See **Defining the Adapter Implementation** on page 6-2 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about defining an Adapter implementation.

### A Note about Import Usage Files

When importing usage data via files, it is generally recommended that the data be split into multiple smaller files instead of importing a single large file. The Adapter Payload Splitter converter can be used to split usage data based on user-defined rules. See **Payload Splitter** on page 6-35 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about using the Adapter Payload Splitter.

## Implementing Import/Usage Services and Business Rules

The Oracle Utilities Meter Data Management application can use a specific type of Oracle Utilities Adapter Business Rule type, called “Import/Usage” for importing usage data in non-standard formats. The Echelon Converter (see **Echelon Converter** on page 5-25) is a pre-configured Business Rule of this type.

The steps outlined below should be performed for **each** “Import/Usage” business rule you wish to run using the Oracle Utilities Adapter.

### Create and Configure Import Global Functions and Versions

The first step in implementing Import/Usage services and business rules is to create and configure any Global Functions used by Import/Usage business rules.

Global Function records for Import/Usage services and business rules **MUST** include the following:

- **Business Rule Type: Import/Usage**

See **Create and Configure Global Functions** on page 6-12 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about creating and configuring Global Functions and Versions.

**Note:** This step need only be performed once, since Global Functions by definition are shared by multiple business rules.

## Create RDLs and RDL Versions

The next step is to create the Rule Description Language (RDL) records and RDL Versions for each business rule.

Rule Description Language records for Import/Usage services and business rules **MUST** include the following:

- **Rule Type Code: Import/Usage**

See **Create and Configure Rule Description Language** on page 6-13 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about creating RDLs and RDL Versions.

## Configure RDLs

The next step is to configure each RDL Version that will be used. For Import/Usage RDLs, this involves the following:

- Mapping data elements/attributes to Meter Data Management tables/columns
- Configuring Table Validations
- Configuring Mappings
- Validating RDL

See **RDL Configuration** on page 6-14 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about configuring RDLs.

## Create Business Rule Record

The first step is to create the Business Rule record(s) that represent the specific business rules.

Business Rule records for “Import/Usage” services and business rules **MUST** include the following:

- **Rule Type: Import / Usage**

See **Create and Configure Business Rule Records** on page 6-16 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about creating Business Rule records.

## Define Business Rule Properties and Parameters

As part of creating the Business Rule record, you must also define the Business Rule properties and parameters

“Import/Usage” business rules use the following properties:

- **xmlstream:** Instructs the Adapter parsing engine to keep the XML data that is to be sent to the database engine
- **mode:** Instructs the Adapter parsing engine to perform database operations on a per table basis (“event”) or as a batch done at the end of the incoming payload document (“batch”).

### Example:

```
<PROPERTIES>
  <PROPERTY NAME='xmlstream' VALUE='n' />
  <PROPERTY NAME='mode' VALUE='event' />
</PROPERTIES>
```

See **Defining Properties and Parameters for Adapter Business Rules** on page 11-107 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about business rule properties and parameters.

## Associate Business Rule to RDL

The next step is to associate each Business Rule record to the corresponding RDL. See **Associating Business Rules to Rule Description Language** on page 11-61 in the *Oracle Utilities Energy Information Platform User's Guide* for more information.

## Create Runtime Service

The next step is to create a Runtime Service record for each business rule. See **Create and Configure Runtime Services** on page 6-19 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information.

## Define Service Properties

As part of creating the Runtime Service record, you must also define the Service properties. See **Service Properties** on page 6-20 in the *Oracle Utilities Energy Information Platform Configuration Guide* and **Defining Service Properties for Adapter Runtime Services** on page 11-28 in the *Oracle Utilities Energy Information Platform User's Guide* for more information.

## Create Service Activation record

The last step is to create a Service Activation record for each Runtime Service. See **Create Service Activation Records** on page 6-21 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information.

## Import/Usage Business Rule Example

**Business Requirement:** Import usage data from external format from the file system for use with Oracle Utilities Meter Data Management.

### About the Payload Data

In this example, the payload data is in a non-standard XML format.

### RDL Record

Create a Rule Description Language record as follows:

- **RDL Name:** MDM\_Ext\_Import
- **Rule Type Code:** Import/Usage

### RDL Version Record

Create an RDL Version record as follows:

- **RDL Name:** MDM\_Ext\_Import
- **Start Time:** MM/DD/YYYY mm:HH:ss
- **Stop Time:** MM/DD/YYYY mm:HH:ss
- **Note:** Version 1 (or an applicable note)

### Business Rule Record

Create a Business Rule record as follows:

- **Rule Name:** MDM\_Ext\_Import
- **Description:** Import Meter Data - MDM
- **Rule Type:** Import/Usage
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME='xmlstream' VALUE='n' />
```

```
<PROPERTY NAME='mode' VALUE='event' />
</PROPERTIES>
```

## Runtime Service Record

Create a Runtime Service record as follows:

- **Name:** MDM\_Ext\_Import
- **JMS Queues:** N/A
- **JMS Filter:** N/A
- **JMS Parameters:** -server -Djava.endorsed.dirs=./lib
- **Origination System:** FILELOADER
- **Parameters:** -N/A
- **Internal Flag:** Y
- **Business Rule Flag:** N
- **Output Flag:** N
- **Runtime Service Type:** FPORTAL
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME="STORAGE_LEVEL" VALUE="2" />
  <PROPERTY NAME="RULE_NAME" VALUE="MDM_Ext_Import" />
  <PROPERTY NAME="RDL_MEMORY_SIZE" VALUE="3" />
  <PROPERTY NAME="POLL_DIRECTORY" VALUE="C:\Import\MDM_EXT" />
  <PROPERTY NAME="POLL_EXTENSION" VALUE="XML" />
  <PROPERTY NAME="POLL_INTERVAL" VALUE="10" />
</PROPERTIES>
```

## Service Activation Record

Create a Service Activation record as follows:

- **Server:** <SERVER\_NAME>
- **Runtime Services:** MDM\_Ext\_Import
- **Enabled:** Yes
- **Run at Startup:** No
- **Keep Alive:** No

## Implementing MDMElsterImport Services and Business Rules

The Oracle Utilities Meter Data Management application uses a specific type of Oracle Utilities Adapter Business Rule type, called “MDMElsterImport” for importing usage data in the Elster format.

The steps outlined below should be performed for **each** “MDMElsterImport” business rule you wish to run using the Oracle Utilities Adapter.

### Create Business Rule Record

The first step is to create the Business Rule record(s) that represent the specific business rules.

Business Rule records for “MDMElsterImport” services and business rules **MUST** include the following:

- **Rule Type:** MDMElsterImport



See **Create and Configure Business Rule Records** on page 6-16 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about creating Business Rule records.

### Define Business Rule Properties and Parameters

As part of creating the Business Rule record, you must also define the Business Rule properties and parameters

“MDMElsterImport” business rules use the following properties:

- **xmlstream:** Instructs the Adapter parsing engine to keep the XML data that is to be sent to the database engine
- **mode:** Instructs the Adapter parsing engine to perform database operations on a per table basis (“event”) or as a batch done at the end of the incoming payload document (“batch”).

**Example:**

```
<PROPERTIES>
  <PROPERTY NAME='xmlstream' VALUE='n' />
  <PROPERTY NAME='mode' VALUE='event' />
</PROPERTIES>
```

See **Defining Properties and Parameters for Adapter Business Rules** on page 11-107 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about business rule properties and parameters.

### Create Runtime Service

The next step is to create a Runtime Service record for each business rule. See **Create and Configure Runtime Services** on page 6-19 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information.

### Define Service Properties

As part of creating the Runtime Service record, you must also define the Service properties. See **Service Properties** on page 6-20 in the *Oracle Utilities Energy Information Platform Configuration Guide* and **Defining Service Properties for Adapter Runtime Services** on page 11-28 in the *Oracle Utilities Energy Information Platform User's Guide* for more information.

### Create Service Activation record

The last step is to create a Service Activation record for each Runtime Service. See **Create Service Activation Records** on page 6-21 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information.

## MDMElsterImport Business Rule Example

**Business Requirement:** Import usage data in Elster format from the file system for use with Oracle Utilities Meter Data Management.

### About the Payload Data

MDMElsterImport business rules use data in the Elster format.

### Business Rule Record

Create a Business Rule record as follows:

- **Class Name:** N/A
- **Rule Name:** MDM\_Elster\_Import
- **Description:** Import Meter Data - MDM
- **Rule Type:** MDMElsterImport

## Runtime Service Record

Create a Runtime Service record as follows:

- **Name:** MDM\_Elster\_Import
- **JMS Queues:** N/A
- **JMS Filter:** N/A
- **JMS Parameters:** -server -Djava.endorsed.dirs=../lib
- **Origination System:** FILELOADER
- **Parameters:** -N/A
- **Internal Flag:** Y
- **Business Rule Flag:** N
- **Output Flag:** N
- **Runtime Service Type:** FPORTAL
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME="STORAGE_LEVEL" VALUE="2" />
  <PROPERTY NAME="RULE_NAME" VALUE="MDM_Elster_Import" />
  <PROPERTY NAME="RDL_MEMORY_SIZE" VALUE="3" />
  <PROPERTY NAME="POLL_DIRECTORY" VALUE="C:\Import\Elster_MDM" />
  <PROPERTY NAME="POLL_EXTENSION" VALUE="XML" />
  <PROPERTY NAME="POLL_INTERVAL" VALUE="10" />
</PROPERTIES>
```

## Service Activation Record

Create a Service Activation record as follows:

- **Server:** <SERVER\_NAME>
- **Runtime Services:** MDM\_Elster\_Import
- **Enabled:** Yes
- **Run at Startup:** No
- **Keep Alive:** No

## Implementing MDMReady Services and Business Rules

The Oracle Utilities Meter Data Management application uses a specific type of Oracle Utilities Adapter Business Rule type, called “MDMReady” for importing usage data in (or converted into) the LSM format. Several of the data converters described in **Oracle Utilities Meter Data Management Data Converters** on page 5-17 can be used with Business Rules of this type to import usage.

The steps outlined below should be performed for **each** “MDMReady” business rule you wish to run using the Oracle Utilities Adapter.

### Create Business Rule Record

The first step is to create the Business Rule record(s) that represent the specific business rules.

Business Rule records for “MDMReady” services and business rules **MUST** include the following:

- **Rule Type:** MDMReady

See **Create and Configure Business Rule Records** on page 6-16 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about creating Business Rule records.

## Define Business Rule Properties and Parameters

As part of creating the Business Rule record, you must also define the Business Rule properties and parameters

“MDMReady” business rules use the following properties:

- **xmlstream:** Instructs the Adapter parsing engine to keep the XML data that is to be sent to the database engine
- **mode:** Instructs the Adapter parsing engine to perform database operations on a per table basis (“event”) or as a batch done at the end of the incoming payload document (“batch”).

**Example:**

```
<PROPERTIES>
  <PROPERTY NAME='xmlstream' VALUE='n' />
  <PROPERTY NAME='mode' VALUE='event' />
</PROPERTIES>
```

See **Defining Properties and Parameters for Adapter Business Rules** on page 11-107 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about business rule properties and parameters.

## Create Runtime Service

The next step is to create a Runtime Service record for each business rule. See **Create and Configure Runtime Services** on page 6-19 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information.

## Define Service Properties

As part of creating the Runtime Service record, you must also define the Service properties, including any specific data converters. See **Service Properties** on page 6-20 in the *Oracle Utilities Energy Information Platform Configuration Guide* and **Defining Service Properties for Adapter Runtime Services** on page 11-28 in the *Oracle Utilities Energy Information Platform User's Guide* for more information.

## Create Service Activation record

The last step is to create a Service Activation record for each Runtime Service. See **Create Service Activation Records** on page 6-21 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information.

## MDMReady Business Rule Example - Usage Import

**Business Requirement:** Import usage data in LSM format from the file system for use with Oracle Utilities Meter Data Management.

### About the Payload Data

MDMReady business rules use data in the LSM format. See **Oracle Utilities Meter Data (.LSM) File Format** on page 5-2 for details about this format.

### Business Rule Record

Create a Business Rule record as follows:

- **Class Name:** N/A
- **Rule Name:** MDM\_Import
- **Description:** Import Meter Data - MDM
- **Rule Type:** MDMReady
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME='xmlstream' VALUE='n' />
  <PROPERTY NAME='mode' VALUE='event' />
</PROPERTIES>
```

## Runtime Service Record

Create a Runtime Service record as follows:

- **Name:** MDM\_Import
- **JMS Queues:** N/A
- **JMS Filter:** N/A
- **JMS Parameters:** -server -Djava.endorsed.dirs=./lib
- **Origination System:** FILELOADER
- **Parameters:** -N/A
- **Internal Flag:** Y
- **Business Rule Flag:** N
- **Output Flag:** N
- **Runtime Service Type:** FPORTAL
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME="STORAGE_LEVEL" VALUE="2" />
  <PROPERTY NAME="RULE_NAME" VALUE="MDM_Import" />
  <PROPERTY NAME="RDL_MEMORY_SIZE" VALUE="3" />
  <PROPERTY NAME="POLL_DIRECTORY" VALUE="C:\Import\MDM" />
  <PROPERTY NAME="POLL_EXTENSION" VALUE="LSM" />
  <PROPERTY NAME="POLL_INTERVAL" VALUE="10" />
</PROPERTIES>
```

## Service Activation Record

Create a Service Activation record as follows:

- **Server:** <SERVER\_NAME>
- **Runtime Services:** MDM\_Import
- **Enabled:** Yes
- **Run at Startup:** No
- **Keep Alive:** No

## MDMReady Business Rule Example - Web Transactions

This sample rule is used to execute operations performed by end users of the Oracle Utilities Meter Data Management application. A rule of type MDMReady, executed by a runtime service of type MDMPORTAL **MUST** be running in order for users to upload, save, validate, and/or approve usage readings.

**Note:** The sample Business Rule and Runtime Service records below are pre-populated in the database. Activating this service and rule requires creating a Service Activation record.

## Business Rule Record

Create a Business Rule record as follows:

- **Class Name:** NULL
- **Rule Name:** LSBR\_MDMUSAGE

- **Description:** MDM Usage
- **Rule Type:** MDMReady
- **Time Expire:** 0

### Runtime Service Record

Create a Runtime Service record as follows:

- **Name:** LSMDM\_WebPortal
- **JMS Queues:** N/A
- **JMS Filter:** N/A
- **JMS Parameters:** -server -Djava.endorsed.dirs=./lib
- **Origination System:** LTMH
- **Parameters:** -N/A
- **Internal Flag:** Y
- **Output Flag:** Y
- **Runtime Service Type:** MDMPORTAL

- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME="RULE_NAME" VALUE="LSBR_MDMUSAGE" />
  <PROPERTY NAME="POLL_INTERVAL" VALUE="2" />
</PROPERTIES>
```

### Service Activation Record

Create a Service Activation record as follows:

- **Server:** <SERVER\_NAME>
- **Runtime Services:** LSMDM\_WebPortal
- **Enabled:** Yes
- **Run at Startup:** Yes
- **Keep Alive:** Yes

## Configuring Partial Commit and Error Management of Import Files

When processing large import files with multiple readings, Oracle Utilities Meter Data Management can be configured to support partial commit of readings as well as logging of readings that fail critical validations.

### Partial Commit

When importing multiple usage readings in a single import file, Oracle Utilities Meter Data Management rejects readings that fail critical validations, and commits all readings that pass all critical validations. When this occurs, the inbound status of the import file is set to “Warning” and all exceptions are tracked in the Inbound Exceptions table.

To control how often readings are committed, define the `SERVER_TRANSACTION_SIZE` property for the Runtime Service being used to import the readings. The value of this property defines the number of readings to commit in each server transaction. In the example below, the application will commit every 500 readings.

```
<PROPERTY NAME="SERVER_TRANSACTION_SIZE" VALUE="500" />
```

## Error Management

Oracle Utilities Meter Data Management can be configured to produce files containing readings that have critical exceptions and can not be imported into the Oracle Utilities Data Repository.

Files are created for each payload file processed. The application creates error files for each unique Meter ID, UOM, and Channel ID, or for each MDMUSAGE element within an LSM file. Only readings (METERREAD elements) and events (METEREVENTS and MDMEVENTS element) that are rejected by the application are captured in the error file. Error files are valid LSM files which can be imported via the Adapter or the Usage Add/Edit screen on the Oracle Utilities Meter Data Management user interface.

### Error File Naming Conventions

Error files are named using the following naming convention.

```
[UIDHUBINBOUND]-[READING_METER]-[yyyyMMdd]=[HHmmss]=[ffff]-[RANDOM].lsm
```

where:

- [UIDHUBINBOUND] is the UID of the record on the Inbound (HUBINBOUND) table
- [READING\_METER] is the Meter ID or Serial Number, the Channel ID, and UOM code (Example: Meter1101)
- [yyyyMMdd] is the date on which the file was created, expressed as a 4 digit year + 2 digit month + 2 digit day (Example: 20081031)
- [HHmmss] is the time at which the file was created, expressed as hours (24 hour), minutes, and seconds (Example: 095500)
- [ffff] is fractions of a second represented in 5 digits (Example: 44533)
- [RANDOM] is a random number generated by the application

**Example:** An error file for Meter 1 (Channel ID 1, UOM 01) created by 9:55 AM on October 31 2008 would be named as follows:

```
Meter1101-20081031-095500-44533-54321.lsm
```

### Error File Location

The location where error files are created can be specified by setting the "MDMERRORDIR" attribute in the IMPORT element in the MDM.CFG.XML configuration file, or by setting the "MDM\_ERROR\_DIR" Runtime Service property.

When using the "MDMERRORDIR" attribute in the MDM.CFG.XML file, error files for all import services are created in the same location. When using the "MDM\_ERROR\_DIR" runtime service property, only errors from payloads processed by the service would be captured in the error file.

**Example (MDM.CFG.XML):** Log all errors for all services in the "c:\mdm\errors\" directory.

```
<MDM>
  <IMPORT MDMERRORDIR="c:\mdm\errors" />
</MDM>
```

**Example (Runtime Service):** Log all errors for the service in the "c:\mdm\errors\" directory.

```
<PROPERTY NAME="MDM_ERROR_DIR" VALUE="c:\mdm\errors" />
```

## Oracle Utilities Meter Data Management Adapter Properties

Service Properties define how Adapter Runtime Services function during runtime. Service properties are used to associate a business rule to a service property, to specify the data source(s) that the service will interact with, to specify how the service creates output based on exceptions, and other properties.

When executing Oracle Utilities Meter Data Management business rules via the Adapter, the following properties specify specific behaviors of the application.

### SERVER\_TRANSACTION\_SIZE

- **Required:** No
- **Possible Values:** Any integer
- **Description:** The Adapter can operate in one of two modes when executing Oracle Utilities Meter Data Management business rules: client transaction management or server transaction management. Client transaction management, or client mode, is the default mode, in which case each payload constitutes a single transaction. In server transaction management, or server mode, the Adapter can manage transactions at a more granular level. The value of this property is the number of transactions (i.e. readings) to process prior to committing the readings.

**Note:** Errors can result when filling missing start times and/or values when importing multiple consumption (or TOU) readings that have no start time or start value for the same meter via multiple services. To force data of this type to be processed serially, set this property to a value of '1.'

- **Default Value:** 0
- **Example:**

```
<PROPERTY NAME="SERVER_TRANSACTION_SIZE" VALUE="500" />
```

### MDM\_ERROR\_DIR

- **Required:** No
- **Possible Values:** A valid network path
- **Description:** The path to a directory on the file system where error files are created for critical exceptions.
- **Default Value:** N/A
- **Example:**

```
<PROPERTY NAME="MDM_ERROR_DIR" VALUE="c:\mdm\errors" />
```

# Oracle Utilities Meter Data Management Validations and Estimations

This section describes several types of validations used by Oracle Utilities Meter Data Management, such as replacement reading validations, out of order reading validations, register reading validations, as well as how it performs estimation operations on partially and entirely missing usage readings. This section includes:

- **Critical Validations**
- **Conditional Validations**
- **Validations and Unit of Measure**
- **Replacement Reading Validations**
- **Out of Order Validations**
- **Register Readings and Validations**
- **Meter Changeout Validations**
- **When and How Estimation is Performed**
- **Interval Estimations**
- **Consumption Estimations**
- **Setting Up Automated Estimation Processing**

## Critical Validations

Critical validations are validations that check that usage readings contains valid data, such valid meter IDs, channel IDs and UOM codes. Critical validations include database referential integrity checks, and other checks that ensure that only valid data enters the system. Note that in this context, the term “valid data” does not necessarily mean that the readings will pass all the validations defined for the meter.

Readings that fail one or more critical validations are not imported into the database, the Inbound record for the payload is flagged as “Error,” and corresponding records will be created in the Inbound Error table.

**Note:** If using the Partial Commit option, failed readings are written to an error file. See **Configuring Partial Commit and Error Management of Import Files** on page 5-41 for more information.

The critical validations performed by Oracle Utilities Meter Data Management include:

<b>Validation</b>	<b>Description</b>
Invalid Meter ID	Compares Meter ID in the payload file to active Meter ID in the database
Invalid UOM	Compares UOM in payload file to UOM in the database for the same Meter ID
Invalid Channel ID	Compares Channel ID in payload file to Channel ID in the database for the same Meter ID



<b>Validation</b>	<b>Description</b>
Too Many Intervals	Examines the number of intervals in (interval readings only) the payload file based on Start Time, Stop Time, and SPI. Can be overridden by Interval Data validation or INTDCONFIG.CFG.XML
Too Few Intervals	Examines the number of intervals in (interval readings only) the payload file based on Start Time, Stop Time, and SPI. Can be overridden by Interval Data validation or INTDCONFIG.CFG.XML
Invalid Start Time or Stop Time	Verifies that the Start Time and Stop Time are valid date/time values.
Stop Time greater than Start Time	Validates that the Stop Time is greater than the Start Time
Invalid Seconds-Per-Interval (SPI)	Compares the intervals in the payload file to ensure they match the SPI in the payload file.
Usage for inactive meter	Checks to determine if usage is for a time period either prior to the Start Time or after the Stop Time of the Meter Configuration record for the meter.
Valid XML	Checks for valid XML
Valid SPI	Validates that the SPI in the payload file is valid

## Conditional Validations

Conditional validations are validations that are automatically executed when certain conditions exist. Conditional validations cannot be disabled as long as the triggering conditions are present. When usage reading fail conditional validations, they trigger exceptions and set the Usage Category of the reading to “Staging” and the Usage Status to “Failed.” The exception code for exceptions triggered by conditional validations cannot be changed.

The conditional validations performed by Oracle Utilities Meter Data Management include:

Validation	Description
Maximum Read Value	<p>Validates that the STARTVAL and STOPVAL values of the reading don't exceed the maximum possible value based on the Number of Dials value specified in either the usage reading or the Meter Configuration record for the meter.</p> <p>Condition: This validation is automatically performed if the Number of Dials is specified in either the usage reading (NUMDIALS attribute of the METERREAD element in the LSM format) or in the Meter Configuration record for the meter.</p>

## Validations and Unit of Measure

All Oracle Utilities Meter Data Management validations check that the UOM for the reading matches the user-defined Unit of Measure. If the UOM for the reading does not match that specified for the validation, the validation is not executed. If a Unit of Measure is not supplied for a validation, the validation is applied to readings of all UOMs.

## Replacement Reading Validations

Replacements readings are defined incoming readings whose Start Time and/or Stop Time overlap in some way with existing readings. This section outlines how the Oracle Utilities Meter Data Management application can be configured to handle replacement readings.

**Note:** The replacement reading requirements and rules outlined in this section apply to interval readings **ONLY**.

### Replacement Reading Requirements

Oracle Utilities Meter Data Management can be configured to support the following requirements:

- Readings are not automatically rejected if they overlap with one or more readings that already exist.
- Ability to generate an informational exception when a new reading overlaps or completely replaces an existing reading.
- Ability to reject data if it would replace manually edited data. (Configurable: can be turned on or off)
- Replacement readings should not cause data to be lost. If a reading exists in the database for 2/1 through 2/28 and a new record comes in for the same meter for 2/1 through 2/15, the data for the two readings will be merged.
- The Reading Category for fully replaced data, that is, data that has the same Start Time as an existing reading, will be set to INACTIVE when newer data is received
- Open Usage Exceptions and Work Queue items for a reading will be closed if/when the reading is completely replaced or set to INACTIVE. If a reading is partially replaced, open Work Queue items and Exceptions remain open.

- Informational exceptions (those with an Exception Type of INFORMATIONAL) are not closed and they remain viewable from within the Oracle Utilities Meter Data Management user interface.

## Configuring Replacement Reading Rules

Rules for handling replacement interval readings can be configured at the system level via the MDM.CFG.XML configuration file, or at the meter level via Validation Groups.

### Configuring System Level Rules

Configuring rules for replacement readings at the system level involves modifying the MDM.CFG.XML configuration file to include the REPLACEMENTOVERLAP attribute. Possible settings for this attribute include:

- **ALLOW**: Use the rules outlined under **Replacement Reading Requirements** on page 5-46
- **REJECT**: Reject replacement readings unless it meets the following criteria:
  - The Start Time and Stop Time for the incoming reading exactly match a reading already in the database
  - The new reading matches exactly one and only one existing reading
  - If the new reading does not match exactly one cut, then it must be replacing the most recent reading for the meter. For example, if the last reading for a meter is from 2/1/06 through 2/28/06, if a new reading came in for 2/1/06 through 3/30/06, it would not be rejected.
- **REJECTMAN**: Use the rules outlined under **Replacement Reading Requirements** on page 5-46, unless the reading being replaced has been manually edited, in which case the incoming reading will be rejected.

### Configuring Meter Level Rules

Configuring rules for replacement readings at the meter level involves defining specific validations to check for replacement readings. These validations include:

- **Replacement Flag** (MDM->All Rules)
 

Replacement Flag validations flag usage as replaced if the reading Start Time or Stop Time overlap with a previous reading in any way.

See **Replacement Flag** on page 3-20 in the *Oracle Utilities Meter Data Management User's Guide* for more information about this validation.
- **Partial Replacement Flag** (MDM->Interval Rules)
 

Partial Replacement Flag validations allow you to specify the Exception Code that will be generated if a partial replacement occurs. The Exception Type (ERROR or INFORMATIONAL) for the Exception Code for the validation (see below) determines whether the Reading Category of overlapping readings are set to STAGING or FINAL. Partial replacements are only allowed if the Overlap parameter on the Replacement validation (see below) is set to **Allow All**.

See **Partial Replacement Flag** on page 3-42 in the *Oracle Utilities Meter Data Management User's Guide* for more information about this validation.
- **Replacement** (MDM->Interval Rules)
 

Replacement validations define how the Oracle Utilities Meter Data Management system will handle replacement readings. A replacement reading is any reading whose Read Start Time and/or Read Stop Time overlap in any way with an existing reading for the same meter.

  - **Allow All**: Allow replacement reads, including partial replacements, based on the rules outlined under **Replacement Reading Requirements** on page 5-46. The Replacement

Flag and Partial Replacement Flag validations dictate the type of exception generated by replacement reads.

- **Reject All:** Reject replacement readings unless it meets the following criteria:
  - The Start Time and Stop Time for the incoming reading exactly match a reading already in the database
  - The new reading matches exactly one and only one existing reading
  - If the new reading does not match exactly one cut, then it must be replacing the most recent reading for the meter. For example, if the last reading for a meter is from 2/1/06 through 2/28/06, if a new reading came in for 2/1/06 through 3/30/06, it would not be rejected.
- **Reject Manual:** Use the rules outlined under **Replacement Reading Requirements** on page 5-46, unless the reading being replaced has been manually edited, in which case the incoming reading will be rejected.

See **Replacement** on page 3-42 in the *Oracle Utilities Meter Data Management User's Guide* for more information about this validation.

## Out of Order Validations

Out of Order validations allow for consumption readings to be received “out of order,” for situations when a reading comes in that overlaps with a reading already in the system. When this validation is enabled, the Start Time and Start Reading of any out of order readings will be adjusted based on the Stop Time and Stop Reading of existing readings in the system.

For example, in the following scenario, the third (3) reading will be corrected.

Out of order reading imported:

Reading	Start Time	Implied Value based in previous reading	Stop Time	Value
1	01/01/07 1:00	9800	01/02/07 1:00	10000
2	01/02/07 1:01	10000	01/03/07 1:00	12000
3	01/02/07 1:01	10000*	01/04/07 1:00	14000

\* Bad start reading and date/time

Out of order reading is imported, Start Time and Start Reading is corrected:

Reading	Start Time	Implied Value based in previous reading	Stop Time	Value
1	01/01/07 1:00	9800	01/02/07 1:00	10000*
2	01/02/07 1:01	10000	01/03/07 1:00	12000
3	01/03/07 1:01	12000**	01/04/07 1:00	14000

\*\* Corrected Start reading and values

In the above example, the validation will trigger an exception for the third reading stating "the start date/time and values were adjusted because an out of order read was received." The system also marks any corrected usage as replaced.

If this validation is not configured, any out of order readings are rejected.

## Configuring Out of Order Validations

Rules for handling out of order readings can be defined at the meter level via Validation Groups.

Configuring rules for out of order readings at the meter level involves including the Out of Order validation within validation groups used for consumption readings, and specifying the OUTFORDERADJUSTMENT exception code, and the unit of measure for which the validation (and adjustment) is performed.

## Register Readings and Validations

Some interval meter data collection systems collect register readings (Start Reading and Stop Reading) separate from interval values for the same reading. When Oracle Utilities Meter Data Management is used with these systems, interval meter readings can come in two segments: a register segment (containing register readings but no interval values), and an interval segment (containing interval values but no register readings).

This section outlines the rules for how Oracle Utilities Meter Data Management handles accepting interval readings in interval and register segments.

### Importing Reading Segments

Interval reading segments can be imported in the same manner as complete interval readings.

- Interval segments are treated as normal interval readings, and validated as normal.
- Register segments are flagged as register readings, and validated, though Oracle Utilities Meter Data Management can be configured such that certain interval validations be skipped for register readings.

### Merging Segments

Oracle Utilities Meter Data Management can be configured to not only accept separate interval and register segments, but also automatically merge segments once both segments have been imported. The Register Reading validation specifies whether or not to merge segments. See **Configuring Register Reading Rules** on page 5-49 for more information.

### Merging Rules

Segments can only be merged if the Start Time and Stop Time of the reading segments match exactly.

The manner in which segments are merged differs slightly depending on the order in which the segments arrive in the Oracle Utilities Meter Data Management system.

When an interval segment comes in first:

- The interval reading is imported and validated as normal.
- When the corresponding register segment is imported, the two segments are merged, and then the entire reading is re-validated.

When a register segment comes in first:

- The reading is imported, flagged as a register reading, and validated as normal (note that some interval validations can be configured to not run for register segments, see **Configuring Register Reading Rules** on page 5-49 for more information).
- When the corresponding interval segment is imported, the two segments are merged, and then the entire (combined) reading is re-validated.

### Configuring Register Reading Rules

Rules for handling interval readings in interval and register segments can be defined at the meter level via Validation Groups.

Configuring rules for register readings at the meter level involves including the Register Reading validation within validation groups used for interval readings, and specifying whether or not to include certain validations for register readings.

### Register Reading Validations

Register Reading validations estimate interval values for interval readings that contain register readings only. This validation also specifies whether or not the application should attempt to merge register segments and interval segments (readings containing interval values but not register readings). See **Register Reading** on page 3-47 in the *Oracle Utilities Meter Data Management User's Guide* for more information about this validation.

### Applying Validations to Register Readings

The **Include when Register Segment** parameter allows you to specify whether or not the following validations are performed on register segments:

- **High/Low Interval** (MDM→Interval Rules)

High/Low Interval validations compare cumulative consumption for the meter to historical data for the same period during the previous year (if not available, use the previous month).

See **High/Low Interval** on page 3-43 in the *Oracle Utilities Meter Data Management User's Guide* for more information about this validation.

- **Energy Sum Check 2** (MDM→Interval Rules)

Energy Sum Check 2 validations check the difference between the energy use defined by consumption values (based on interval meter register readings) and the energy use recorded in the intervals to a multiple of the meter multiplier (typically 2). The validation uses the Meter Multiplier supplied in the reading, or the Meter Multiplier defined in the Meter Configuration record for the meter.

See **Energy Sum Check 2** on page 3-45 in the *Oracle Utilities Meter Data Management User's Guide* for more information about this validation.

- **Zero Energy from Intervals** (MDM→Interval Rules)

Zero Energy from Intervals validations check for zero usage from the sum of the interval values of the current meter reading.

See **Zero Energy from Intervals** on page 3-46 in the *Oracle Utilities Meter Data Management User's Guide* for more information about this validation.

- **KVARH Check** (MDM→Interval Rules)

KVARH Check validations identify intervals where reactive load (kVARh) is present and active load (kWh) is not, indicating a suspicious usage pattern and possible meter malfunction. This validation is only performed if kVARh data is available.

See **KVARH Check** on page 3-46 in the *Oracle Utilities Meter Data Management User's Guide* for more information about this validation.

## Meter Changeout Validations

Overtime, the associations between physical and logical meters can change. This can occur when a physical meter is damaged or replaced. When the current physical meter that is installed at a service point is swapped out for a new physical meter, it results in a "Meter Changeout." In this instance, the existing logical data channel (Meter Data Channel) may still exist, but a new physical device is capturing usage information. This is represented in Oracle Utilities Meter Data Management by the creation of a new Meter Configuration record for the logical meter that specifies the new physical meter.

Oracle Utilities Meter Data Management includes validations that can check for meter changeouts, check if readings overlap a meter changeout, and can calculate the start time and value for

consumption and TOU readings that occur on or near meter changeout and that are missing the start time and/or start value.

## Checking for Meter Changeouts

Oracle Utilities Meter Data Management checks for meter changeouts by checking the Meter Configuration table for the meter. If there is more than one Meter Configuration record for the physical meter, a meter changeout has occurred. Both the **Meter Changeout Overlap Check** and the **Meter Changeout Start Time and Value** validations can be used to check for meter changeouts.

## Calculating Start Time and Value

When a meter changeout is detected for a reading, Oracle Utilities Meter Data Management needs to take the new physical meter into account when calculating the start time and value for consumption and TOU meter readings that are missing the start time and/or start value. The start time for a meter changeout reading can be based on the Stop Time of the previous reading (plus one second) for the meter or on the Start Time of the most recent Meter Configuration record. The start value for a meter changeout reading can be based on the Stop Value of the previous reading for the meter or can be set to zero (0). The Meter Changeout Start Time and Value validation (available under Consumption and TOU validations) can be used to calculate the start time and value consumption and TOU readings. The **Meter Changeout Start Time and Value** validation can be used to calculate these values.

The following example illustrates how start time and value are calculated.

## Sample Database Records

In this example, a meter changeout occurs at January 14, 2009 at midnight (00:00:00), as outlined by the following Meter Configuration records:

Meter Configuration	Start Time	Stop Time
1	01/01/2009 00:00:00	01/13/2009 23:59:59
2	01/14/2009 00:00:00	

A previous reading for the meter exists from January 1, 2009 through January 15, 2009, with a Stop Value of 200, and a new reading comes in for the meter with a Stop Time of January 31, 2009 and a Stop Value of 300. The new reading is missing both the start time and start value. This is illustrated by the following meter readings:

Reading	Start Time	Start Value	Stop Time	Stop Value
1	01/01/2009 00:00:00	100	01/15/2009 23:59:59	200
2			01/31/2009 23:59:59	300

## Calculating Start Time

The two options for calculating the start time of the second reading are as follows:

- **Use Previous Stop Time** (sets the Start Time of the reading equal to the Stop Time of the previous meter reading plus one second)
- **Use Meter Configuration Record:** Uses the Start Time of the most recent Meter Configuration record for the meter.

If the Meter Changeout Start Time and Value validation is configured with the **Use Previous Stop Time** option, the Start Time of the second reading would be 01/16/2009 00:00:00

If the Meter Changeout Start Time and Value validation is configured with the **Use Meter Configuration Record** option, the Start Time of the second reading would be 01/14/2009 00:00:00.

### Calculating Start Value

The two options for calculating the start value of the second reading are as follows:

- **Use Previous Stop Value.** Sets the Start Value of the reading equal to the Stop Value of the previous meter reading.
- **Use Zero:** Set the Start Value of the reading to zero (0).

If the Meter Changeout Start Time and Value validation is configured with the **Use Previous Stop Value** option, the Start Value of the second reading would be 200.

If the Meter Changeout Start Time and Value validation is configured with the **Use Zero** option, the Start Value of the second reading would be 0.

### Checking for Meter Changeout Overlaps

Oracle Utilities Meter Data Management checks for meter changeout overlaps by comparing the start and stop time of the usage period with the current Meter Configuration record for the meter. If more than one physical meter is active during the duration of the usage period, a meter changeout overlap has occurred. The **Meter Changeout Overlap Check** validation can be used to create exceptions when meter changeout overlaps occur.

### Configuring Meter Changeout Rules

Rules for handling meter changeouts can be defined at the meter level via Validation Groups.

Configuring rules for meter changeouts at the meter level involves including the Meter Changeout Overlap Check validation within validation groups used for all types of meters, and the Meter Changeout Start Time and Value validation within validation groups used for consumptions and TOU readings.

## When and How Estimation is Performed

There are two instances where estimations are performed.

The first instance only applies to interval data and is when there is a gap within an interval cut that is being loaded. For example, the cut is for 01/01/2005 00:00:00 through 01/31/2005 23:59:59 and there is data missing for 01/15/2005 00:00:00 - 01/15/2005 23:59:59. This type of estimation is called Interval Estimation, and is performed upon import of data if the Interval Gap or Interval Estimations validation is defined.

The second instance applies to both interval and consumption data and is when there are missing reads based on the read cycle associated to the meter.

Estimation of missing reads is performed via the MDM Estimations command line program. See **Setting Up Automated Estimation Processing** on page 5-57 for more information.

## Interval Estimations

Interval estimations are those performed on interval readings. This section details how Oracle Utilities Meter Data Management performs interval estimations, including:

- **Interval Estimation Parameters**
- **Types of Interval Estimations**
- **Interpolation**
- **Averaging**



## Interval Estimation Parameters

The following estimation parameters are configured as part of the Interval Gap and/or Interval Estimation Rules validations:

- **Average:** A flag that indicates (Yes or No) whether or not averages of intervals should be used to estimate missing values
- **Days to Average:** The number of days to average when using averages to estimate missing data
- **User Prior Cuts Only:** A flag that indicates (Yes or No) whether or not to limit the intervals used in averaging estimation calculations should come only from prior to the intervals being estimated. The default is "yes". When set to "No", the estimation rule can use interval values from a later date than the intervals being estimated.
- **Interpolation:** A flag that indicates (Yes or No) whether or not linear interpolation should be used to estimate missing values
- **Maximum Hours:** The maximum number of hours to be interpolated when both Average and Interpolation are set to yes.
- **Exception Code:** The exception code for the MDM exception created if/when interval estimations are performed
- **Unit of Measure:** The unit of measure for which interval estimations are performed

## Types of Interval Estimations

If an interval gap occurs, estimation rules are executed as the interval data is loaded. There are two different approaches to estimation, Interpolation and Averaging. The specific method used is based on the parameters defined for the validation.

- If only the Interpolation flag is set to "YES", Interpolation is used to estimate the entire gap.
- If only the Average flag is set to "YES", Averaging is used estimate the entire gap.
- If both the Average and Interpolation flags are set to "YES" and the gap is less the Maximum Hours value, Interpolation is used to estimate the gap.
- If both the Average and Interpolation flags are set to "YES" and the gap is more the Maximum Hours value, Interpolation is used to estimate up to the Maximum Hours, and Averaging is used to estimate the remaining missing intervals.

**Note:** All estimated intervals have a Status Code of "M."

## Interpolation

Interpolation performs linear interpolation to estimate missing interval values, and is used when the Interpolation flag is set to "YES" in the Interval Estimation validation.

## Terminology

Below are some specific terms used in the descriptions of the interval estimation calculations performed by Oracle Utilities Meter Data Management.

- **Estimation Adder:** A calculated value used to estimate interval values with interpolation estimation.
- **Flat Load:** An estimation in which all missing values are assigned the same estimated value
- **Left Value:** the value of the interval immediately preceding an interval gap
- **Like Day:** Using a similar "type" of day. a weekday (NOT a holiday) for a missing weekday, a weekend day for a missing weekend day
- **Right Value:** the value of the interval immediately following an interval gap
- **Same Weekday:** Using the same weekday (e.g. using Mondays for missing Mondays, etc.)

The interpolation estimation process differs depending on where the interval gap exists:

### **For Gaps In the Middle of a Cut**

When the interval gap falls in the middle of an interval reading, the interpolation process is as follows:

- Get the value of the interval immediately preceding the gap (the “Left Value”).
- Get the value of the interval immediately after the gap (the “Right Value”).
- Subtract the Left Value from the Right Value, divide result by number of missing intervals in the gap plus one. The result is called the Estimation Adder.
- For each interval in the gap, add the Estimation Adder to the interval immediately before the one being estimated.

#### **Example:**

If the intervals in a cut are as follows:

10, <missing>, <missing>, 25

The Left Value would be 10, the Right Value would be 25. The Estimation Adder would be 5.

$$[(25 - 10) / (2 + 1) = 5]$$

The resulting estimated cut would be as follows:

10, **15**, **20**, 25

### **For Gaps At the Beginning of a Cut**

When the interval gap falls at the beginning an interval reading, the interpolation process determines if there is valid historical data for the interval immediately preceding the start of this cut. If there is, it uses it as the “Left Value” and uses the rules outlined under **For Gaps In the Middle of a Cut** on page 5-54. If there is no valid historical data for the interval immediately preceding the gap, it estimates a “Flat Load” for the gap by assigning the “Right Value” (the value of the interval immediately following the interval gap) to all intervals in the gap.

### **For Gaps Which Extend to the End of a Cut**

When the interval gap falls at the end an interval reading, the interpolation process estimates a “Flat Load” for the gap by assigning the “Left Value” (the value of the interval immediately preceding the interval gap) to all intervals in the gap.

## **Averaging**

By default, average estimations perform the following for each missing interval:

- First, the system attempts to use intervals from “Same Weekdays” and averages their values.
- If that fails, the system use intervals from “Like Days” and averages their values.
- Missing holidays require slightly different logic, in which the system tries to using matching holidays within the current and historical period. If none are available, the system substitutes missing Holidays with Sundays.

The general rule is to use data from days that are closest in time to the missing interval. If two days are equidistant from a missing day, the earlier one is to be used.

By default the system uses only historical interval data (intervals from a point in time prior to the interval being estimated). However, if the "Use Prior Cuts Only" parameter is set to "No", the estimation logic can also use "future" intervals (those from a point in time after the interval being estimated).

Note that historical interval data used in estimations must have a Reading Category of FINAL. In addition, the interval values used in estimation calculations are selected based on the ESTCOMPVALUE and ESTCOMPOPERATION attributes in the MDM.CFG.XML

configuration file. By default, interval value with any status code are used in estimations. The ESTCOMPVALUE and ESTCOMPOPERATION attributes can be used to filter the interval values used in estimation calculations based on status code. See **MDM.CFG.XML Configuration File** on page 3-5 for more information about these attributes.

### Estimating Missing Intervals That Fall on a Holiday

The system determines if the missing interval is during a holiday. If the missing interval falls during a holiday, it then determines if there is another holiday later in the reading.

- If there is a holiday later within the reading and it is usable, the system gets the value from the same time on this day as the missing interval's start time and adds it to the values to average.
- If there isn't a holiday later in the reading, the system scans back day by day, for up to 90 days, from the start of the missing interval to find a holiday and performs the same usability check when one is found. If this holiday is usable, OK, the system gets the value from the same time on this day as the missing interval's start time and adds it to the values to average.
- If there isn't a usable holiday within the current reading or 90 days back from the start of the missing holiday interval, the system then checks the following in order:
  - 1st attempt: The Sunday before the missing holiday
  - 2nd attempt: The first Sunday after the missing holiday interval
  - 3rd attempt: Two Sundays before
  - 4th attempt: Two Sundays after
  - 5th attempt: Three Sundays before
  - 6th attempt: Three Sundays after

The system continues this pattern until reaching the ends of potential values or a number of intervals equal to the Intervals to Average parameter (or the default of 3) have been found, whichever comes first. The system averages all values found and substitutes the result for the missing interval.

### Estimating Missing Intervals That Fall on Weekdays or Weekend Days

If the missing interval is NOT during a holiday, the system first uses the "Same Weekday" method, and checks the following in order:

- 1st attempt: The "same day" (e.g. Monday for Monday) in the week before the missing interval
- 2nd attempt: The "same day" in the week after the missing interval
- 3rd attempt: The "same day" 2 weeks before
- 4th attempt: The "same day" 2 weeks after
- 5th attempt: The "same day" 3 weeks before
- 6th attempt: The "same day" 3 weeks after

The system continues this pattern until reaching the ends of potential values or a number of intervals equal to the Intervals to Average parameter (or the default of 3) have been found, whichever comes first. The system averages all values found and substitutes the result for the missing interval.

If there is not a single usable "Same Weekday" interval, the system uses the "Like Day" method, and checks the following in order:

- 1st attempt: The "Like Day" (e.g. weekday [non-holiday] for weekday) before the missing day
- 2nd attempt: The "Like Day" after
- 3rd attempt: Two "Like Days" before

- 4th attempt: Two “Like Days” after

The system continues to alternate before and after until reaching the ends of potential values or a number of intervals equal to the Intervals to Average parameter (or the default of 3) have been found, whichever comes first. The system averages all values found and substitutes the result for the missing interval.

## Consumption Estimations

Consumption estimations are those performed on consumption and Time Of Use (TOU) readings. This section details how Oracle Utilities Meter Data Management performs consumption estimations, including:

- **Consumption Estimation Parameters**
- **Types of Consumption Estimation**

### Consumption Estimation Parameters

The following estimation parameters are configured as part of the Consumption Estimation Rules validation:

- **Days:** The number of days of historical data required
- **Exception Code:** The exception code for the MDM exception created if/when consumption estimations are performed
- **Unit of Measure:** The unit of measure for which consumption estimations are performed

### Types of Consumption Estimation

Two methods to estimate usage are provided, depending on availability of historical data.

#### Estimation Based on Historical Data

If billing data is available for the same meter for the previous year (and it was not estimated), the system calculates the Average Daily Usage (ADU) for the previous year using the then-current read cycle and uses this value as the ADU. To determine the read cycle for the previous year, the system takes the read cycle being estimated and subtracts one year from the start and stop times. For example, if the estimated read cycle is 01/01/2005 - 01/31/2005 then the historical read cycle would be 01/01/2004 - 01/31/2004.

#### Estimation Based on Previous Month

If there is no data from the previous year but there is a full preceding billing month (of at least a number of days equal to the Days parameter or the default value of 27), the system calculates the ADU for the preceding billing month and uses this value for the ADU. In this case, the estimation algorithm is estimation based on preceding billing period's data.

#### Calculating Consumption

To calculate the estimated consumption, the system multiplies the estimated ADU (including any constants or factors) by the number of days since the last good reading. If necessary, the system divides this value by a meter constant or other factor to convert it to the same units reported in the meter reading. The system rounds using the number of decimals specified for the meter.

Note that historical consumption data used in estimations must have a Reading Category of FINAL, and must not have been estimated (Estimated Flag = No).

## Setting Up Automated Estimation Processing

In the case of missing reads, estimations are performed via the MDM Estimations command line program. This program is typically configured as a daily batch process to locate and estimate missing reads based on read cycles.

The MDM Estimations command line program (MDMEstimations.EXE) uses the syntax shown below. Parameter switches are case insensitive (i.e. you can enter them in either upper or lower case (-c or -C)).

**MDMEstimations.exe** -d <connectstring> -q <qualifier> [-gap] [-gather] [-days] [-tolerance] [-utc] [-count] [-id]

The command must be entered on one line. Also, you must either change to the directory in which the program is stored (typically, \LODESTAR\bin) before entering the command, or specify the path in the command. To view a list of all parameters on-screen, type **MDMEstimations -?** at the command prompt.

Parameter	Description
-d	<p><i>connectstring</i> is database connection information for the Oracle Utilities Data Repository that contains the rate form record. This parameter is <b>required</b> and must be in one of the following formats:</p> <p>For Oracle databases:</p> <pre>"Data Source=&lt;data_source&gt;;User ID=&lt;user_id&gt;;Password=&lt;password&gt;;LSProvider=ODP;"</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• &lt;data_source&gt; is the Oracle TNS Name for the data source, from the TNS_NAMES.ora file (typically located in the \\&lt;machine&gt;\oracle\network\admin directory)</li> <li>• &lt;user_id&gt; is the user ID for the database connection</li> <li>• &lt;password&gt; is the password for the supplied user ID.</li> </ul>
-q	<i>qualifier</i> is the qualifier for the data source.
-gap	Runs the program in gap-checking mode. If not present, the program runs in missing read mode.
-gather	Executes a stored procedure that gathers the appropriate data needed to perform estimations.
-days	Overrides the DAYS attribute in the MDM.CFG.XML configuration file. Used only during the “gather” step of the estimation process. See <b>Defining Estimation Rules</b> on page 5-58 for more information.
-tolerance	Overrides the TOLERANCE attribute in the MDM.CFG.XML configuration file. Used only during the “gather” step of the estimation process. See <b>Defining Estimation Rules</b> on page 5-58 for more information.
-utc	Specifies the type of usage to estimate. Valid values include “INTERVAL” and “CONSUMPTION.” If not specified, usage for both Interval and Consumption meters is estimated.
-count	The number of processes to run. Used to run multiple instances of the program on the same machine. See <b>Running Multiple Instances of MDMEstimations</b> on page 5-59 for more information.

Parameter	Description
-id	The number of the current process. Used to run multiple instances of the program on the same machine. See <b>Running Multiple Instances of MDMEstimations</b> on page 5-59 for more information.

The MDM Estimation program creates a log file in the C:\LODESTAR\Log directory called "MDMEstimations\_<DSN>\_<Qualifier>.log, where:

- <DSN> is the DSN specified with the -d parameter
- <Qualifier> is the qualifier specified with the -q parameter

The MDM Estimation program uses the Read Cycle associated with the Meter to determine when the meter should be read. For example, if a meter is associated to a Daily Read Cycle then it is expected to receive a read for each day of the year.

The MDM Estimation program is run in two distinct modes: data gathering and estimation processing.

- **Data Gathering** (-gather): Used to gather the data needed to perform estimations.
- **Estimation Processing**: Used to perform estimation calculations

## Defining Estimation Rules

The missing read query the uses ESTIMATIONS element in the MDM.CFG.XML configuration file to define rules for estimations performed. This program scans for meters with missing reads based on Read Cycles, based on the following attributes specified in this element.

- **DAYS**: The number of days in the past the system should check when identifying missing reads to estimate. For example, a setting of 60 would mean that the MDMEstimations.exe program would check the past 60 days for missing reads. The default value is 30.
- **TOLERANCE**: The number of days before a reading is considered "late." For example, a setting of 5 would estimate readings for meters whose Read Cycles have Read Cycle Dates that are more than 5 days old, and would ignore meters whose Read Cycle Dates are within the last 5 days. The default value is 5.
- **GAPSTARTDAYSBACK**: The number of days before the current date to use as the start date when checking for gaps. Used to define a range of days used when checking for gaps.
- **GAPSTOPDAYSBACK**: The number of days before the current date to use as the stop date when checking for gaps. Used to define a range of days used when checking for gaps.

A missing read can be either an entire read cycle or just a portion of the read cycle. For example, if the read cycle is monthly, you could be missing the entire month or just a portion of the month.

Depending on the type of usage, interval or consumption, different estimation approaches are used by the MDM Estimations program. For interval estimations:

- Interpolation estimation is used for gaps that are less than the Maximum Hours value (or the default of 2 hours). See **Interpolation** on page 5-53 for more information.
- Averaging estimation is used for gaps that are more than the Maximum Hours value (or the default of 2 hours). See **Averaging** on page 5-54 for more information.

## Examples

Run in "gathering" mode for missing reads:

```
MDMEstimations.exe {db params omitted} -gather
```

Run in "gathering" mode for gap checking:

```
MDMEstimations.exe {db params omitted} -gather -gap
```

Run missing read validations (after gathering has been completed):

```
MDMEstimations.exe {db params omitted}
```

Run gap check validations (after gap gathering has been completed)

```
MDMEstimations.exe {db params omitted} -gap
```

## Running Multiple Instances of MDMEstimations

The MDM Estimations command line program can be configured to run as multiple instances when running in Estimation Processing mode.

The “-count” and “-id” command line parameters are used to allow multiple instances. The first specifies the number of instances to run, and the second specifies the ID of the instance. This method allows multiple processes to be run on multiple application servers.

Sample command line for 4 total instances might look like this:

```
MDMEstimations.exe {db params omitted} -id 1 -count 4
```

```
MDMEstimations.exe {db params omitted} -id 2 -count 4
```

```
MDMEstimations.exe {db params omitted} -id 3 -count 4
```

```
MDMEstimations.exe {db params omitted} -id 4 -count 4
```





# Chapter 6

---

## Setting Up Oracle Utilities Meter Data Management Billing Determinant Calculations

This chapter describes how to set up Oracle Utilities Meter Data Management billing determinant calculations, including:

- **Overview**
- **Understanding Bill Determinant Calculations**
- **Configuring Bill Determinant Calculations**

## Overview

The Oracle Utilities Meter Data Management bill determinant functionality provides a framework for calculating bill determinant values for accounts, service points, and meters based on usage stored in the Oracle Utilities Data Repository and calculations defined in Oracle Utilities Rules Language. While calculations are performed at the account-level (meaning that calculations are executed for accounts), bill determinant values can be calculated and saved at three levels: account-level, service point-level, and meter-level.

### The Bill Determinant Calculation Process

At a high-level, the bill determinant calculation process includes the following steps:

1. A bill determinant calculation request is initiated. Calculation requests can be initiated through the user interface, or via batch or a web service.
2. The **Account Selector** creates a record in the **BD Request** table for the request, and creates records in the **BD Queue** table for each eligible account.
3. An Adapter service picks up each queue record and triggers the **Calculator Engine** to execute calculations for the account. Calculation rules are defined in the Oracle Utilities Rules Language.
4. Once calculations are complete, the record in the **BD Queue** table is moved into the **BD Queue Archive** table and its status is updated.

**Note:** If using the Calculator Engine in Publisher mode, BD Queue records are not moved into the BD Queue Archive table until the bill determinants have been published.

# Understanding Bill Determinant Calculations

In order to understand how Oracle Utilities Meter Data Management bill determinant calculations work, it's important to understand the components involved. This section describes the components of the bill determinant calculation functionality, including:

- **Database Tables**
- **User Interface**
- **AccountSelector Web Service**
- **AccountSelector Interface**
- **Account Selector**
- **Adapter Queue**
- **Adapter Services and Business Rules**
- **Calculator Engine**

## Database Tables

Oracle Utilities Meter Data Management bill determinant calculations use a number of tables in the Oracle Utilities Data Repository. This section provides brief descriptions of these tables and the role they play in the bill determinant calculation process. For more information about these tables, see **Setting Up Oracle Utilities Meter Data Management Bill Determinant Calculation Data** on page 4-20.

**Note:** Many of the tables used with Oracle Utilities Meter Data Management billing determinant calculations are also used by Oracle Utilities Billing Component. If both Oracle Utilities Meter Data Management and Oracle Utilities Billing Component are licensed, it is recommended that only one of the products be used for all billing calculations.

### Account-Related Tables

Account-related tables define the accounts, service points, and meters for which bill determinants can be calculated. These tables also define the cycles in which bill determinants are calculated, and the business rules used to perform those calculations. Account-related tables include the following:

#### Account

Records in the Account table represent accounts for which billing determinants are calculated. Accounts are linked to customers in the **Customer** table, and to service points in the **Service Point Accounts** table.

#### Account History

Records in the Account History table store relationships between accounts and billing cycles (defined in the **Billing Cycle** table). These records help Oracle Utilities Meter Data Management determine the correct read (bill) date for an account

#### Bill Determinants

Records in the Bill Determinants table represent types of billing determinants that can be calculated. The actual values for bill determinants are stored in the **Bill History Values**, **Meter Bill Determinant Values**, and **Service Point Bill Determinant Values** tables.

#### Bill History

Records in the Bill History table represent billing periods for which billing determinants are calculated. The results of billing determinant calculations are stored in the **Bill History Values**, **Meter Bill Determinant Values**, and **Service Point Bill Determinant Values**.

### **Billing Cycle**

Records in the Billing Cycle table identify the billing cycles used by operating companies/ jurisdictions. The actual dates for each billing cycle are stored in the **Billing Cycle Dates** table.

### **Billing Cycle Dates**

Records in the Billing Cycle Dates table define the read date for each bill month in a billing cycle.

### **Customer**

Records in the Customer table represent customers which are the parents of accounts (see the **Account** table).

### **Rate Code**

Records in the Rate Code table define codes associated with rate schedules (from the **Rate Form** table) that define the business rules used in billing determinant calculations.

### **Rate Code History**

Records in the Rate Code History table represent relationships between accounts and rate codes (from the **Rate Code** table). Oracle Utilities Meter Data Management uses the records in this table to determine which rate schedules to use when processing billing determinant calculations for an account.

### **Rate Form**

Records in the Rate Form table represent rate schedules that define the business rules used in billing determinant calculations. See **Rate Forms** on page 1-4 in the *Oracle Utilities Rules Language User's Guide* for more information about rate form versions.

### **Rate Form Version**

Records in the Rate Form Version table represent different versions of a rate form defined in the **Rate Form** table. See **Rate Form Versions** on page 1-5 in the *Oracle Utilities Rules Language User's Guide* for more information about rate form versions.

### **Service Point**

Service points are points where service is provided. Bill determinants can be calculated for each service point associated with an account. Service points are linked to accounts in the **Service Point Accounts** table and are linked to meters in the **MDM Meter Table** table.

### **Service Point Accounts**

Records in the Service Point Accounts represent relationships between accounts (from the **Account** table) and service points (from the **Service Point** table).

### **Run Time Tables**

Run time tables capture instances of bill determinant calculation requests and processes, and include the following:

#### **BD Queue Type**

Records in the BD Queue Type table define specific queues used when initiating billing determinant calculation requests.

#### **BD Queue**

Records in the BD Queue table represent individual billing determinant calculation processes (one for each account) that are currently being executed. Each BD Queues record will have one related record in the **BD Request** table.

**BD Queue Archive**

Records in the BD Queue Archive table represent individual billing determinant calculation processes that have been previously executed. Each BD Queue Archive record will have one related record in the **BD Request** table.

**BD Request**

Records in the BD Request table represent individual billing determinant calculation requests. Each BD Request record will have one or more related records in the **BD Queue** or **BD Queue Archive** tables.

**Bill Determinant Calculation Results Tables**

Bill determinant calculation results tables store the results of bill determinant calculations for accounts, service points, and meters. These tables include the following:

**Bill History Values**

The Bill History Values is used to store account-level billing determinant calculation results for a billing period in the **Bill History** table.

**Bill View**

The Bill View table is used store billing determinant calculation results for a billing period in the **Bill History** table. Results are stored in XML format.

**Meter Bill Determinant Values**

The Meter Bill Determinant Values table is used to store meter-level billing determinant calculation results for a billing period in the **Bill History** table.

**Service Point Bill Determinant Values**

The Service Point Bill Determinant Values table is used to store service point-level billing determinant calculation results for a billing period in the **Bill History** table.

**User Interface**

The Oracle Utilities Meter Data Management includes user interface screens used to initiate calculations and search and view calculation request and results, including:

- **Initiate Billing Determinant Generation**
- **BD Requests**
- **Results and Values**
- **Scheduled Jobs**

**Initiate Billing Determinant Generation**

The **Initiate Billing Determinant Generation** screen is used to initiate ad-hoc billing determinant generation for selected accounts. See **Initiating Billing Determinant Calculations** on page 8-2 in the *Oracle Utilities Meter Data Management User's Guide* for more information.

**Note:** This feature requires that the BD\_SELECTOR\_TRIGGER Adapter service be properly configured. See **Configuring Adapter Services** on page 6-20 for more information about configuring this service.

**BD Requests**

Whenever billing determinants are calculated, a record is inserted into the BD Request table to record the process, and a record for each account is inserted into the BD Queues table to record the calculation process for that account. The **BD Requests** screen is used to search for and view billing determinant calculation requests and queue records. See **Searching and Viewing Billing**

**Determinant Calculation Requests** on page 8-3 in the *Oracle Utilities Meter Data Management User's Guide* for more information.

## Results and Values

The billing periods for which billing determinants are calculated for each account are defined in the Bill History table. Billing determinant calculation results are stored in the Bill History Value table (account-level billing determinants), the Service Point Bill Determinant Values table (service point-level billing determinants), and the Meter Bill Determinant Values (meter-level billing determinants) table. The **Results and Values** feature can be used to search for and view billing determinant calculation results. See **Searching and Viewing Billing Determinant Results and Values** on page 8-6 in the *Oracle Utilities Meter Data Management User's Guide* for more information.

## Scheduled Jobs

Billing determinant calculation jobs can be scheduled from the Initiate Billing Determinant Generation screen. The **Scheduled Jobs** screen can be used to search for and view scheduled billing determinant calculation requests. See **Viewing Scheduled Billing Determinant Calculation Jobs** on page 8-11 in the *Oracle Utilities Meter Data Management User's Guide* for more information.

## Account Selector

When billing determinant calculations are initiated, the account selector creates records in the **BD Request** table and identifies accounts that are eligible for billing determinant calculations. The account select can be triggered via the user interface, the Account Selector web service, or the AccountSelector (COM) interface.

## Input Parameters

The account selector takes the following input parameters:

- **Accounts to process:** The specific accounts for which bill determinants are to be calculated. Bill determinant calculations can be initiated for All accounts, a specific Account ID, or a list of Account IDs.
- **Read Date:** The date for which the bill determinants are to be calculated. This date is used by the account selector to determine eligible accounts, and by the **Calculator Engine** to calculate the start and stop of the billing period.
- **Queue Type:** The queue type used in the **BD Queue** and **BD Queue Archive** table. See **Adapter Queue** on page 6-11 for more information. The Queue Type parameter can only be specified when initiating calculations using the user interface. The Account Selector web service and COM interface do NOT support specifying the queue type.

For each account identified as eligible, the account selector inserts a record into the **BD Queue** table to record the calculation process for the account.

## Determining Account Eligibility

An account is considered “eligible” for billing determinant calculations if the following conditions are true:

- The account is active as of the specified Read Date. Accounts are considered active if the Stop Time on the Account table is NULL or set to a date later than 35 days from the current date. If the Stop Time is not NULL, the account is eligible until there is a Bill History record with a Stop Time greater than or equal to the account Stop Time.
- The Billing Mode flag for the account is set to “Fully Automatic.”
- The specified Read Date falls within the account’s eligibility window.

An account's eligibility window for each bill month is based on its Read Date for the bill month (see **Account Read Dates** on page 6-7) and the Pre and Post Windows at both the default and account level. If not set at the account level, the default Pre and Post Windows are used.

For example, consider an account with a Read Date of March 8 for the March bill month. If the Pre and Post Windows were not set for the account, it would use the default Pre and Post Windows (set to 3 and 5 respectively), and the account's eligibility window would be March 5 (March 8 minus 3 days for the Pre Window) through March 13 (March 8 plus 5 days for the Post Window). In this case, the account would be considered eligible for any specified Read Date that falls between March 5 and March 13. However, if the account's Pre and Post Windows were set to 3 and 3, the account's eligibility window would be March 5 (March 8 minus 3 days for the Pre Window) through March 11 (March 8 plus 3 days for the Post Window), and the account would be considered eligible for any specified Read Date that falls between March 5 and March 11.

- The account has a Rate Code History record in effect for the specified Read Date. A Rate Code History record is considered in effect if the specified Read Date falls between the Start Time and Stop Time on the Rate Code History record.
- If there is a Bill History record for the account for the bill month, the Bill Time must be Null.
- There are no open work queue items of Work Queue Types where Stop Billing is set to "Automatic Only" or "Both." Accounts that have open work queue items are considered ineligible for bill determinant calculations until any open work queue items are closed.

### Account Read Dates

An account's read dates are determined from the Billing Cycle associated to the account in the Account History table. Each Billing Cycle has a set of associated read dates and bill months defined in the Billing Cycle Dates table.

### Creating BD Queue and BD Queue Archive Records

For each account processed, the account selector creates a record in either the **BD Queue** or **BD Queue Archive** table.

- When supplied a specific account ID or a list of accounts, the account selector creates a record in BD Queue table for each eligible account, and creates a record in the BD Queue Archive table for each non-eligible (or duplicate) account. In some cases, non-eligible accounts also create work queue items. See **Error Conditions and Results** for additional information about errors and work queue items created by the account selector.
- When the account selector is run for "All" accounts, it creates BD Queue records only for eligible accounts. Records for non-eligible accounts are NOT created in the BD Queues Archive table.

### Default Billing Options

When the account selector runs, it uses a pre-defined user called "MDMBD." This "user" uses a set of pre-defined Default Billing Options. The calculator engine also run as the MDMBD user, and uses these settings. See **Default Billing Options** on page 6-15 for more information about these pre-defined options.

### Recalculation Requests

The Account Selector can also process requests for recalculation of bill determinants based on replacement usage. These requests use different input parameters, as follows:

- **Meter ID:** The logical meter ID for which replacement usage has been received.
- **Start and Stop Time:** The start and stop time of the replacement usage.

When the Account Selector processes a recalculation request, it looks up the account ID related to the supplied meter ID, determines the Read Date the recalculation process, sets the Bill Time of the corresponding Bill History record to Null, and creates a record for the request in the **BD Queue** table.

Recalculation requests are only enabled if the Bill Determinant Recalculation validation has been configured for the account's meters. See **Bill Determinant Recalculation** on page 3-24 in the *Oracle Utilities Meter Data Management User's Guide* for more information.

### External Calculation Requests

The Account Selector can also process requests for calculation of bill determinants from an external system, such as a customer information system (CIS). External calculation requests use a specific mode of the Account Selector (RunCalcUsage), differ from other requests in a number of ways, outlined below./

#### Parameters

External calculation request use the following input parameters:

- **Account ID:** The account id for the request.
- **Start Time:** The start time of the bill period to calculate.
- **Stop Time:** The stop time of the bill period to calculate.
- **Date Breaks:** A list of dates between the Start Time and Stop Time on which the account was subject to a rate change. When date breaks are supplied in a request, the Account Selector creates a separate Bill History record for each date break.

When the Account Selector processes an external calculation request, it creates Bill History records based on the start time, stop time, and date breaks supplied in the request, and creates a record for the request in the **BD Queue** table.

#### Account Validation

When processing external calculation requests, the Account Selector checks the following:

- The account exists and is active
- The account has a Rate Code History record in effect for the calculation period.
- The account has at least one related service point.

#### Creating Bill History Records for External Calculations Requests

When creating Bill History records for external calculation requests, the Account Selector uses the following guidelines:

- The start time of the first Bill History records created for external calculation requests is equal to the start time of the request. If there no date breaks, the stop time of the Bill History record is equal to the stop time of the request, and no additional records are created.
- If a date break is provided in the request, the stop time of the first Bill History record is set to 1 second before the first date break, and an additional Bill History record is created with a start time equal to the date break and a stop time equal to the stop time in the request.
- If more than one date break is provided in the request, additional Bill History records are created for each date break as described above until the entire period between the supplied start time and stop time is accounted for.
- All Bill History records for external calculation requests have a Read Date equal to the stop time of the request.

For example, if a request contained the followin dates:

- Account ID: 123456



- Start Time: 02/01/2010 00:00:00
- Stop Time: 02/28/2010 23:59:59
- Date Break: 02/15/2010 00:00:00

the Account Selector would create the following Bill History records:

Account ID	Bill Start	Bill Stop	Read Date
123456	02/01/2010 00:00:00	02/14/2010 23:59:59	02/28/2010 23:59:59
123456	02/15/2010 00:00:00	02/28/2010 23:59:59	02/28/2010 23:59:59

Any existing Bill History records for the calculation periods are deleted if the Bill Time is not NULL and the Published flag is set to “Yes.”

If the Bill Time is NULL or the Publisher flag is set to “No” on an existing Bill History record, a record is created in the BD Queue Archive table is created with a status of “PUBLISHED..”

See **Processing Bill Determinant Requests** on page B-20 for more information about processing external calculation requests.

## Error Conditions and Results

The table below summarizes the most common error conditions for non-eligible accounts and the expected results:

Condition	Result
Inactive Account	BD Queue Archive <ul style="list-style-type: none"> <li>• BD Status Code: Error</li> <li>• Business Status Code: POSTED</li> <li>• Business Message: Primary check failed</li> </ul>
Invalid Billing Mode Flag	BD Queue Archive <ul style="list-style-type: none"> <li>• BD Status Code: Error</li> <li>• Business Status Code: POSTED</li> <li>• Business Message: Primary check failed</li> </ul>
No active Billing Cycle for account	BD Queue Archive <ul style="list-style-type: none"> <li>• BD Status Code: Error</li> <li>• Business Status Code: POSTED</li> <li>• Business Message: Primary check failed</li> </ul> Work Queue Item <ul style="list-style-type: none"> <li>• Work Queue: MDMBD</li> <li>• Work Queue Type: WARNING</li> <li>• Opened Note: No billing cycle dates found for the account</li> </ul>

Condition	Result
Specified Read Date falls outside the account's eligibility window	BD Queue Archive <ul style="list-style-type: none"> <li>• BD Status Code: Error</li> <li>• Business Status Code: POSTED</li> <li>• Business Message: Primary check failed</li> </ul> Work Queue Item <ul style="list-style-type: none"> <li>• Work Queue: MDMBD</li> <li>• Work Queue Type: WARNING</li> <li>• Opened Note: No billing cycle dates found for the account</li> </ul>
No Rate Code History records exists for account for bill period	BD Queue Archive <ul style="list-style-type: none"> <li>• BD Status Code: Error</li> <li>• Business Status Code: POSTED</li> <li>• Business Message: Rate schedule was not found</li> </ul>
Non-Null Bill Time on Bill History record	BD Queue Archive <ul style="list-style-type: none"> <li>• BD Status Code: Error</li> <li>• Business Status Code: POSTED</li> <li>• Business Message: Account already billed</li> </ul>
Open work queue item (where Stop Billing = "Automatic Only" or "Both") on account	BD Queue Archive <ul style="list-style-type: none"> <li>• BD Status Code: Error</li> <li>• Business Status Code: POSTED</li> <li>• Business Message: WQ item detected</li> </ul>
Duplicate request (a record already exists in the BD Queue table for the account)	BD Queue Archive <ul style="list-style-type: none"> <li>• BD Status Code: Error</li> <li>• Business Status Code: DUPLICATE</li> <li>• Business Message: All checks completed</li> </ul>
Account does not exist (for external calculation requests)	Work Queue Item <ul style="list-style-type: none"> <li>• Work Queue: MDMBD</li> <li>• Work Queue Type: MDMBD_SYS_N_ACCOUNT</li> <li>• Opened Note: Account does not exist.</li> </ul>

## AccountSelector Web Service

The account selector includes can be invoked via the BDWS web service. See **Executing Bill Determinant Generation using the BDWS Web Service** on page 6-46 for more information about using this web service. The **BDWS** web service includes the following three operations:

### BDCalcAccount

This operation is used to initiate bill determinant calculations for a specific account and uses the following parameters:

- **Input:** UIDACCOUNT for the account to process and a specified Read Date
- **Output:** UID of the Payload Extension record created by the request

### BDCalcAccountAll

This operation is used to initiate bill determinant calculations for all accounts and uses the following parameters:

- **Input:** A specified Read Date
- **Output:** UID of the Payload Extension record created by the request

### BDCalcAccountList

This operation is used to initiate bill determinant calculations for a list of accounts and uses the following parameters:

- **Input:** Name of the account list to process and a specified Read Date
- **Output:** UID of the Payload Extension record created by the request

## AccountSelector Interface

The account selector can also be invoked via the AccountSelector interface, which includes the following two methods:

- **Run:** Used to initiate bill determinant calculations for a specific account, all accounts, or a list of specific accounts
- **RunRecalc:** Used to initiate re-calculations for a specific account.
- **RunCalcUsage:** Used to initiate bill determinant calculations for a specific account based on billing dates specified in the request. This interface is used when integrating Oracle Utilities Meter Data Management with a customer information system (CIS) such as Oracle Utilities Customer Care and Billing.

See **Executing Bill Determinant Calculations using the AccountSelector Interface** on page 6-40 for more information about using the AccountSelector interface.

## Adapter Queue

The **BD Queue** and **BD Queue Archive** are tables used to capture bill determinant calculation processes for each account. When calculations for an account are initiated, a record is created in the **BD Queue** table. As calculations for an account are completed, the BD Queue record is moved into the **BD Queue Archive** table and its status updated.

### BD Queue Types

Each record in both the BD Queue and BD Queue Archive table have a Queue Type column used to further distinguish records in those tables. The value of the Queue Type column on each queue record is based on the Queue Type parameter specified in the calculation request.

Adapter services can be configured to process only calculations of a specific queue type. This means that processing of calculations can be spread among multiple different Adapter services and servers. This also means that specific calculations can be directed to specific services to help

prioritize how calculations are processed. For example, in a configuration with five active services, four services might be configured to process calculations in the “Standard Priority” (NORMAL) queue, while the fifth service is configured to process calculations in the “Urgent Business Priority” (URGENT) queue. In this scenario, the four “Standard Priority” service would handle the bulk of the calculation processing, and the “Urgent Business Priority” could be devoted only to urgent processes.

## Adapter Services and Business Rules

Oracle Utilities Meter Data Management uses Energy Information Platform Adapter services and business rules to trigger bill determinant calculations. The descriptions of these services and rules include default settings for Runtime Service properties and Business Rule properties. See **Defining Service Properties for Adapter Runtime Services** on page 11-28 in the *Oracle Utilities Energy Information Platform User’s Guide* for more information about runtime service properties, and **Defining Properties and Parameters for Adapter Business Rules** on page 11-107 in the *Oracle Utilities Energy Information Platform User’s Guide* for more information about business rule properties.

**Note:** Multiple copies of each service can be active at the same time, to spread account processing across servers and services. See **Configuring Adapter Services** on page 6-20 for more information.

### BD\_SELECTOR\_TRIGGER

When calculations are initiated via the user interface or web service, a record is created in the Payload Extension table with a Payload Type of “BDCALC.” The BD\_SELECTOR\_TRIGGER service monitors this table, and when a record is detected, the BD\_SELECTOR\_TRIGGER business rule triggers the account selector to create a record in the BD Request table and records in the BD Queue / BD Queue Archive tables for each account. The BD\_SELECTOR\_TRIGGER uses the following Runtime Service and Business Rule.

#### BD\_SELECTOR\_TRIGGER - Runtime Service

The BD\_SELECTOR\_TRIGGER Runtime Service is defined as follows:

- **Name:** BD\_SELECTOR\_TRIGGER
- **JMS Queues:** N/A
- **JMS Filter:** N/A
- **JMS Parameters:** -server -Djava.endorsed.dirs=./lib
- **Origination System:** ADAPTER
- **Parameters:** N/A
- **Internal Flag:** Y
- **Output Flag:** N
- **Runtime Service Type:** DPORTAL
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME="STORAGE_LEVEL" VALUE="0" />
  <PROPERTY NAME="RULE_NAME" VALUE="BD_SELECTOR_TRIGGER" />
  <PROPERTY NAME="PAYLOAD_TYPE" VALUE="BDCALC" />
  <PROPERTY NAME="PAYLOAD_FORMAT" VALUE="C" />
</PROPERTIES>
```

#### BD\_SELECTOR\_TRIGGER - Business Rule

The BD\_SELECTOR\_TRIGGER Business Rule is defined as follows:

- **Rule Name:** BD\_SELECTOR\_TRIGGER

- **Description:** Fires the BD selector with the data in the payload.
- **Rule Type:** COM
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME="method" VALUE="RunCalc" />
  <PROPERTY NAME="progid" VALUE="Lodestar.AccountSelector" />
</PROPERTIES>
```

## BD\_RECALCULATE\_SELECTOR\_TRIGGER

When replacement usage is detected for an account for which bill determinants have previously been calculated, a record is created in the Payload Extension table with a Payload Type of “BDRECALC.”. The BD\_RECALCULATE\_SELECTOR\_TRIGGER service monitors this table, and when a record is detected, the BD\_RECALCULATE\_SELECTOR\_TRIGGER business rule triggers the account selector to create a record in the BD Request table and records in the **BD Queue** table for each eligible account. This service is only invoked if the Bill Determinant Recalculation validation has been configured for the account’s meters. See **Bill Determinant Recalculation** on page 3-24 in the *Oracle Utilities Meter Data Management User’s Guide* for more information. The BD\_RECALCULATE\_SELECTOR\_TRIGGER uses the following Runtime Service and Business Rule.

### BD\_RECALCULATE\_SELECTOR\_TRIGGER - Runtime Service

The BD\_RECALCULATE\_SELECTOR\_TRIGGER Runtime Service is defined as follows:

- **Name:** BD\_RECALCULATE\_SELECTOR\_TRIGGER
- **JMS Queues:** N/A
- **JMS Filter:** N/A
- **JMS Parameters:** -server -Djava.endorsed.dirs=./lib
- **Origination System:** ADAPTER
- **Parameters:** N/A
- **Internal Flag:** Y
- **Output Flag:** N
- **Runtime Service Type:** DPORTAL
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME="STORAGE_LEVEL" VALUE="0" />
  <PROPERTY NAME="RULE_NAME" VALUE="BD_RECALCULATE_SELECTOR_TRIGGER" />
  <PROPERTY NAME="PAYLOAD_TYPE" VALUE="BDRECALC" />
  <PROPERTY NAME="PAYLOAD_FORMAT" VALUE="C" />
</PROPERTIES>
```

### BD\_RECALCULATE\_SELECTOR\_TRIGGER - Business Rule

The BD\_RECALCULATE\_SELECTOR\_TRIGGER Business Rule is defined as follows:

- **Rule Name:** BD\_RECALCULATE\_SELECTOR\_TRIGGER
- **Description:** Fires the BD Recalculate selector with the data in the payload.
- **Rule Type:** COM
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME="method" VALUE="ReCalc" />
  <PROPERTY NAME="progid" VALUE="Lodestar.AccountSelector" />
</PROPERTIES>
```

## BD\_QUEUEPOLLER

The BD\_QUEUEPOLLER service monitors the **BD Queue** table and when a record is detected, triggers the **Calculator Engine** for each record. The BD\_QUEUEPOLLER uses the following Runtime Service.

### BD\_QUEUEPOLLER - Runtime Service

The BD\_QUEUEPOLLER Runtime Service is defined as follows:

- **Name:** BD\_QUEUEPOLLER
- **JMS Queues:** N/A
- **JMS Filter:** N/A
- **JMS Parameters:** -server -Djava.endorsed.dirs=./lib
- **Origination System:** ADAPTER
- **Parameters:** N/A
- **Internal Flag:** Y
- **Output Flag:** N
- **Runtime Service Type:** BDQUEUE

- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME="POLL_INTERVAL" VALUE="30" />
  <PROPERTY NAME="POLL_COUNT" VALUE="50" />
  <PROPERTY NAME="QUEUE_TYPE" VALUE="NORMAL" />
  <PROPERTY NAME="RATE_CFG_FILE" VALUE="C:\Lodestar\cfg\Lodestar.cfg" />
</PROPERTIES>
```

**Note:** These are the default properties for this service. These can be changed as needed based on your requirements. See **Configuring Adapter Services** on page 6-20 for more information.

## Calculator Engine

When the BD\_QUEUEPOLLER service detects a record in the BD Queue table, it triggers the calculator engine which performs the actual bill determinant calculations for the account. For each account, the calculator engine does the following:

- Calculates the bill period Start and Stop
- Verifies that there is sufficient usage for the account for the entire bill period
- Executes the rate schedule associated with the account via the current Rate Code History record in effect for the account

Each of these steps is described in more detail below.

### Calculating the Bill Period Start and Stop Time

The calculator engine uses the following rules when calculating the bill period Start and Stop Time:

- If a Bill History is present for the account for the Bill Month (where the Bill Time is Null), use its Start and Stop Times.
- If no Bill History is present, calculate the start time and stop time as follows:
  - **Start Time:** The Stop Time of the most recent Bill History record (where the Bill Time is Not Null) plus one second. If no previous Bill History record can be found, use the Start Time of the account.
  - **Stop Time:** For accounts with consumption and time of use meters, the end of the day (23:59:59) of the Read Date. For accounts with interval meters, the stop time of the last interval reading that includes the Read Date. If an account has both interval and consumption or time of use meters, use the rule for accounts with interval meters.

### Verifying Usage

When verifying usage for the bill period, the calculator checks that there is Final usage (where the Usage Category is “FINAL”) for the account (based on the account’s service points and their related meters) that covers the entire bill period. This means that there is at least one reading for each of the account’s meters with a Stop Time greater than or equal to the Stop Time of the bill period. If there is not sufficient usage for the bill period, an error is returned (see **Error Conditions and Results** on page 6-17).

### Executing Rules Language

The calculator engine executes the rate schedule that is in effect for the account on the specified Read Date, based on the Effective Date on the account’s Rate Code History record.

### Default Billing Options

When the calculator engine runs, it uses a pre-defined “user” called “MDMBD.” This “user” uses a set of pre-defined Default Billing Options. The account selector also runs as the MDMBD “user” and uses these settings. The setting for these options are as follows:

Billing Options Tab	Settings
Billing Options	<ul style="list-style-type: none"> <li>• <b>Default Billing Mode:</b> Automatic Billing</li> <li>• <b>Default Eligibility Pre Window:</b> 3</li> <li>• <b>Default Eligibility Post Window:</b> 5</li> </ul>
Billing Rules	<ul style="list-style-type: none"> <li>• <b>Effective Date is:</b> Bill Stop</li> <li>• <b>Allow Saves without SAVE TO CIS Statement:</b> On (checked)</li> </ul>

Billing Options Tab	Settings
Report Options	<ul style="list-style-type: none"> <li>• <b>Print Detail:</b> Normal (Bill determinant calculations produce no output report)</li> </ul>
Summary Options	<ul style="list-style-type: none"> <li>• <b>Compute Multiple Rates Per Account:</b> No</li> </ul>
Multiple Bill History Records	<ul style="list-style-type: none"> <li>• <b>Allow Multiple Bill History Records per Bill Period/Read Date:</b> No</li> </ul> <p><b>Note:</b> When executing external calculation requests, this should be set as follows:</p> <ul style="list-style-type: none"> <li>• <b>Allow Multiple Bill History Records per Bill Period/Read Date:</b> Bill Individually - Allow Out of Order Bill Periods.</li> </ul>
Check Options	<ul style="list-style-type: none"> <li>• <b>Do not check for Interval Data gaps:</b> On (checked)</li> <li>• <b>Do not check Account Notes:</b> On (checked)</li> <li>• <b>Do not check Override Ready:</b> On (checked)</li> <li>• <b>Do not check Account's Meter Values:</b> On (checked)</li> <li>• <b>Do not check Rate Code Flags:</b> On (checked)</li> <li>• <b>Do not check Rate Schedule Flags:</b> On (checked)</li> <li>• <b>Do not display "Reason for Change" dialog when updating the Bill History [Value] records:</b> On (checked)</li> <li>• <b>Do not check Bill Determinant Use:</b> On (checked)</li> <li>• <b>Use MDM Schema:</b> On (checked)</li> <li>• <b>Use earliest unbilled bill history:</b> Off (unchecked)</li> </ul>

See **Default Billing Options** on page 10-11 in the *Data Manager User's Guide* for more information about Default Billing Options.

### Changing the Default Billing Options

Changing or overriding these settings requires logging on to the Energy Information Platform user interface as the MDMBD user and making the desired changes on the Default Billing Options dialog.

### Additional Settings

In addition to the above billing options, bill determinant calculations use the following pre-defined settings:

#### Save/Approve Options:

- Automatically save/approve each group of related pages if all OK

This setting specifies that saving to the database and output files is enabled for all bill determinant calculation processing.

**Note:** If a Warning occurs during bill determinant calculation for an account, any calculated values will be not saved.

#### Rules Language Interval Data Error Handling:

- Stop with Error

This setting specifies that any error related to interval data will stop processing.



See **Error Handling** on page 2-14 in the Data Manager User’s Guide for more information about Rules Language error handling.

### LODESTAR.CFG Settings

When processing external calculation requests, include the following parameter in the LODESTAR.CFG configuration file on all application servers that will be running the calculator engine:

AUTO\_BILL\_NO\_BILL\_CYCLE = 1

If this line is present, checking for a Billing Cycle and Billing Cycle Date when processing billing for accounts is disabled. Accounts must still have an active Account History record in place, but Billing Cycle information is not needed (and ignored if present). This option should **ONLY** be used in conjunction with on-demand bill determinant calculations initiated via integration between Oracle Utilities Meter Data Management and a customer information system such as Oracle Utilities Customer Care and Billing. See **Appendix B: Integrating Oracle Utilities Meter Data Manager with a Customer Information System** for more information.

### Error Conditions and Results

The table below summarizes the most common error conditions during bill determinant calculations and the expected results

Condition	Error
Valid Rate Code History record, but no active Rate Form Version	BD Queue Archive <ul style="list-style-type: none"> <li>• BD Status Code: Error</li> <li>• Business Status Code: POSTED</li> <li>• Business Message: Rate schedule was not found</li> </ul>
Valid Rate Code History and Rate Form Version, but rate form is empty (i.e. contains no executable statements)	BD Queue Archive <ul style="list-style-type: none"> <li>• BD Status Code: Completed</li> <li>• Business Status Code: FAIL</li> <li>• Business Message: Rate Form Version source text is empty or has no executable statements. Check rate form source and RCDATA path.</li> </ul>
Insufficient usage for the billing period	BD Queue Archive <ul style="list-style-type: none"> <li>• BD Status Code: Completed</li> <li>• Business Status Code: FAIL</li> <li>• Business Message: Detected incomplete data between the billing start/stop times for these meter data channels:</li> </ul> Work Queue Item <ul style="list-style-type: none"> <li>• Work Queue: MDMBD</li> <li>• Work Queue Type: MDMBD_INSUF_USAGE</li> <li>• Opened Note: Detected incomplete data between the billing start/stop times for these meter data channels:</li> </ul>

<b>Condition</b>	<b>Error</b>
Abort or Warning statement issued during processing	<p>BD Queue Archive</p> <ul style="list-style-type: none"><li>• BD Status Code: Completed</li><li>• Business Status Code: FAIL</li><li>• Business Message: [message issued via ABORT or WARNING statement]</li></ul> <p>Work Queue Item</p> <ul style="list-style-type: none"><li>• Work Queue: MDMBD</li><li>• Work Queue Type: ABORT / WARNING</li><li>• Opened Note: [message issued via ABORT or WARNING statement]</li></ul>

# Configuring Bill Determinant Calculations

This section provides guidelines for configuring Oracle Meter Data Management for bill determinant calculations, including:

- **Setting Up Data**
- **Configuring Adapter Services**
- **Configuring Oracle Utilities Rules Language**
- **Executing Bill Determinant Calculations using the AccountSelector Interface**
- **Executing Bill Determinant Generation using the BDWS Web Service**

## Setting Up Data

The first step in configuring bill determinant calculations is to set up the data that will drive the process in the Oracle Utilities Data Repository. This includes setting up account, service points, and meters, as well as billing cycles, rate schedules and rate codes.

- **Set up Accounts, Service Points, and Meters**
- **Define Bill Determinants**
- **Define Billing Cycles and Dates**
- **Define Rate Forms and Rate Codes**
- **Associate Accounts to Billing Cycles and Rate Codes**
- **Define Billing Periods (optional)**

See **Setting Up Oracle Utilities Meter Data Management Bill Determinant Calculation Data** on page 4-20 for more information about the tables referenced below.

### Set up Accounts, Service Points, and Meters

Accounts, service points, and meters are the entities for which bill determinants can be calculated. Set up of account-related data includes the following:

1. Define accounts in the **Account** table.
  - **Note:** The Pre Window and Post Window determine an account's eligibility window for each Read Date. If these are NULL, the Default Billing Option Pre Window (3) and Post Window (5) are used.
2. Define service points in the **Service Point** table.
3. Define meters and meter configurations in the **MDM Meter** and **Meter Configuration** tables.
  - **Note:** The Billed? column in the Meter Configuration table should be set to "Yes" for all meters that will be used for calculating bill determinants.
4. Associate service points to their corresponding meters in the **MDM Meter** table.
5. Associate service points to their corresponding account in the **Service Point Account** table.
  - **Note:** The Start Time and Stop Time in the Service Point Account table determine the dates during which the service point is associated to the account.

### Define Bill Determinants

You must also define the specific bill determinants you plan to calculate. This includes the following:

1. Define bill determinants in the **Bill Determinant** table.

## Define Billing Cycles and Dates

Billing cycles and dates define the dates on which bill determinants are calculated for each account. Set up of this data includes the following:

1. Define billing cycles in the **Billing Cycles** table.
2. Define dates for each billing cycle in the **Billing Cycle Dates** table.

## Define Rate Forms and Rate Codes

Rate forms and rate codes are used to specify the rate schedules used in bill determinant calculations. Set up of this data includes the following:

1. Define the rate schedules and riders that will be used in bill determinant calculations in the **Rate Form** table.
2. Create a record for each rate schedule in the **Rate Code** table.

## Associate Accounts to Billing Cycles and Rate Codes

Before bill determinants can be calculated for an account, the account must be associated to a billing cycle (to specify the account's Read Dates) and rate code (to specify which rate schedule is used in the calculations). This step includes the following:

1. Associate each account to its applicable billing cycle in the **Account History** table.
  - **Note:** The Effective Date determines when the billing cycle is in effect for the account.
2. Associate each account to its applicable rate code in the **Rate Code History** table.
  - **Note:** The Start Time and Stop Time determine the dates during which the rate code is in effect for the account.

## Define Billing Periods (optional)

Billing periods define the dates for which bill determinants are calculated. Billing periods can either be pre-defined ahead of time or can be calculated as part of the bill determinant calculation process (see **Calculating the Bill Period Start and Stop Time** on page 6-15). This step includes the following

1. Define billing periods for each account in the **Bill History** table (optional).
  - **Note:** Each billing period should be aligned with one of the account's Read Dates. The Read Date and Bill Month in each Bill History record should match those in a record in the Billing Cycle Dates table.

## Configuring Adapter Services

The Adapter services used to trigger and execute calculations are pre-defined in the Oracle Utilities Data Repository, but must be properly configured before bill determinant calculations can be performed. This section outlines the configuration of each of these services. See **Chapter 11: Setting Up and Configuring the Energy Information Platform Adapter** in the *Oracle Utilities Energy Information Platform User's Guide* and **Chapter 6: Setting Up, Configuring, and Running the Energy Information Platform Adapter** in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about configuring the Adapter.

## BD\_SELECTOR\_TRIGGER

When calculations are initiated via the user interface or web service, a record is created in the Payload Extension table with a Payload Type of "BDCALC." The BD\_SELECTOR\_TRIGGER service monitors this table, and when a record is detected, the BD\_SELECTOR\_TRIGGER business rule triggers the account selector to create a record in the BD Request table and records in the BD Queue / BD Queue Archive tables for each account. Configuring the BD\_SELECTOR\_TRIGGER service involves the following:

### Enabling the Service

The BD\_SELECTOR\_TRIGGER service must be enabled in the Service Activation table before it can be used. Use the following procedure to enable the BD\_SELECTOR\_TRIGGER service:

1. Select **Tools and Utilities->Adapter Components->Service Activation**.  
The Service Activation screen opens.
2. Select the BD\_SELECTOR\_TRIGGER service in the Runtime Service field and click **Search**.
3. The BD\_SELECTOR\_TRIGGER record appears on the Service Activation screen.
4. Edit the Server field to specify the server on which you wish the service to run and click **Save**.

### Creating multiple BD\_SELECTOR\_TRIGGER services

When calculating bill determinants for large numbers of accounts, you may wish to create multiple copies of the BD\_SELECTOR\_TRIGGER service to spread the work load across multiple services and servers. Use the following procedure to create duplicates of the BD\_SELECTOR\_TRIGGER service:

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.  
The Runtime Service screen opens.
2. Enter “BD\_SELECTOR\_TRIGGER” in the Name field and click **Search**.  
The BD\_SELECTOR\_TRIGGER record appears on the Runtime Service screen.
3. Edit the Name field (to create a unique name for the copy of the service, such as BD\_SELECTOR\_TRIGGER\_1, 2, etc.) and click **Save As New**.

**Note:** Be sure to enable each BD\_SELECTOR\_TRIGGER service in the Service Activation table as described above.)

### BD\_RECALCULATE\_SELECTOR\_TRIGGER

When replacement usage is detected for an account for which bill determinants have previously been calculated, a record is created in the Payload Extension table with a Payload Type of “BDCALC.” The BD\_RECALCULATE\_SELECTOR\_TRIGGER service monitors this table, and when a record is detected, the BD\_RECALCULATE\_SELECTOR\_TRIGGER business rule triggers the account selector to create a record in the BD Request table and records in the **BD Queue** table for each eligible account. This service is only invoked if the Bill Determinant Recalculation validation has been configured for the account’s meters. See **Bill Determinant Recalculation** on page 3-24 in the *Oracle Utilities Meter Data Management User’s Guide* for more information. Configuring the BD\_RECALCULATE\_SELECTOR\_TRIGGER service involves the following:

#### Enabling the Service

The BD\_RECALCULATE\_SELECTOR\_TRIGGER service must be enabled in the Service Activation table before it can be used. Use the following procedure to enable the BD\_RECALCULATE\_SELECTOR\_TRIGGER service:

1. Select **Tools and Utilities->Adapter Components->Service Activation**.  
The Service Activation screen opens.
2. Select the BD\_RECALCULATE\_SELECTOR\_TRIGGER service in the Runtime Service field and click **Search**.  
The BD\_RECALCULATE\_SELECTOR\_TRIGGER record appears on the Service Activation screen.
3. Edit the Server field to specify the server on which you wish the service to run and click **Save**.

### Creating multiple BD\_RECALCULATE\_SELECTOR\_TRIGGER services

When calculating bill determinants for large numbers of accounts, you may wish to create multiple copies of the BD\_RECALCULATE\_SELECTOR\_TRIGGER service to spread the work load across multiple services and servers. Use the following procedure to create duplicates of the BD\_RECALCULATE\_SELECTOR\_TRIGGER service:

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.

The Runtime Service screen opens.

2. Enter “BD\_RECALCULATE\_SELECTOR\_TRIGGER” in the Name field and click **Search**.

The BD\_RECALCULATE\_SELECTOR\_TRIGGER record appears on the Runtime Service screen.

3. Edit the Name field (to create a unique name for the copy of the service, such as BD\_RECALCULATE\_SELECTOR\_TRIGGER\_1, 2, etc.) and click **Save As New**.

**Note:** Be sure to enable each BD\_RECALCULATE\_SELECTOR\_TRIGGER service in the Service Activation table as described above.

### BD\_QUEUEPOLLER

The BD\_QUEUEPOLLER service monitors the **BD Queue** table and when a record is detected, triggers the **Calculator Engine** for each record. Configuring the BD\_QUEUEPOLLER service involves the following:

#### Enabling the Service

The BD\_QUEUEPOLLER service must be enabled in the Service Activation table before it can be used. Use the following procedure to enable the BD\_QUEUEPOLLER service:

1. Select **Tools and Utilities->Adapter Components->Service Activation**.

The Service Activation screen opens.

2. Select the BD\_QUEUEPOLLER service in the Runtime Service field and click **Search**.

The BD\_QUEUEPOLLER record appears on the Service Activation screen.

3. Edit the Server field to specify the server on which you wish the service to run and click **Save**.

#### Creating multiple BD\_QUEUEPOLLER services

When calculating bill determinants for large numbers of accounts, you may wish to create multiple copies of the BD\_QUEUEPOLLER service to spread the work load across multiple services and servers. Use the following procedure to create duplicates of the BD\_QUEUEPOLLER service:

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.

The Runtime Service screen opens.

2. Enter “BD\_QUEUEPOLLER” in the Name field and click **Search**.

The BD\_QUEUEPOLLER record appears on the Runtime Service screen.

3. Edit the Name field (to create a unique name for the copy of the service, such as BD\_QUEUEPOLLER\_1, 2, etc.) and click **Save As New**.

**Note:** Be sure to enable each BD\_QUEUEPOLLER service in the Service Activation table as described above.

#### Adjusting the Runtime Service properties

The runtime service properties of the BD\_QUEUEPOLLER services can be adjusted to fine-tune the performance of each service. Use the following procedure to adjust the BD\_QUEUEPOLLER service’s properties:

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.

The Runtime Service screen opens.

2. Enter "BD\_QUEUEPOLLER" in the Name field and click **Search**.

The BD\_QUEUEPOLLER record appears on the Runtime Service screen.

3. Edit the Properties field as needed and click **Save**.

Some of the properties used by this service include:

- **POLL\_INTERVAL**: Specifies the number of seconds between polling cycles. The default is 30 seconds. To cause one or more services to poll more or less frequently, lower or raise this number as appropriate.
- **POLL\_COUNT**: Specifies the number of records to lock during each polling cycle. The default is 50. To configure one or more services to lock more or less records, raise or lower this number as appropriate. If this property is set to 0, the service will attempt to lock all of the records available.
- **QUEUE\_TYPE**: Specifies the queue type the service will look for in the BD Queue table. There is no default value for this property. If this property is not specified, the service picks up all BD Queue records, regardless of the Queue Type. To configure one or more services to monitor a specific queue, set this property accordingly. See **BD Queue Types** on page 6-11 for more information.
- **RATE\_CFG\_FILE**: Specifies the path and file name of the LODESTAR.CFG configuration file used by the Calculation Engine. The default is "C:\Lodestar\cfg\lodestar.cfg." To designate one or more services to use an alternative configuration file, set this property to the alternative file.
- **DISABLE\_DELETION**: Specifies that the calculation engine should NOT move the record in the BQ Queue table to the BD Queue Archive table upon completion. The only valid value for this property is "TRUE." This property should be used **ONLY** when using the BD Publisher service as part of an integration between Oracle Utilities Meter Data Management and a customer information system such as Oracle Utilities Customer Care and Billing. See **Appendix B: Integrating Oracle Utilities Meter Data Manager with a Customer Information System** for more information.

## Configuring Oracle Utilities Rules Language

After setting up data and configuring the Adapter service, the next step in configuring bill determinant calculations is to configure the calculation business rules and processes using Oracle Utilities Rules Language. Rules Language calculations are defined in Rate Form Version records (associated with Rate Forms and Rate Codes). See **Appendix A: Setting Up Rate Form Records and Rate Codes** in the *Oracle Utilities Rules Language User's Guide* for more information about creating rate forms, rate codes, and rate form versions. This section outlines a number of guidelines for defining bill determinant calculations using Oracle Utilities Rules Language.

### Pre-Defined Identifiers

When executing the Rules Language, the calculator engine calculates and sets a number of pre-defined identifiers that can be accessed in processing. For example, bill determinant calculation rules often reference the BILL\_START and BILL\_STOP identifiers. See **Identifiers** on page 4-2 in the *Oracle Utilities Rules Language User's Guide* for more information about pre-defined identifiers.

The table below lists the pre-defined identifiers available to bill determinant calculations. Identifiers marked with an asterisk (\*) can be assigned within the Rules Language.

Identifier	Description
ACCOUNT.[tail]	The entire account record, available as database identifiers, with a stem of ACCOUNT. See <b>Database Identifiers</b> on page 4-5 in the <i>Oracle Utilities Rules Language User's Guide</i> for more information about database identifiers.
BILL_PERIOD	The first day in the bill month.
BILL_PERIOD_SELECT*	Determines how a season is assigned to a bill period. Its default value of 0 means the system checks the BILL_STOP date against the season dates. If set to 1, the system uses the BILL_START date. If set to 2, the system uses the BILL_PERIOD (bill month) date. If set to 3, the system uses the Scheduled Read Date. If set to 4, the system uses the Governing Date. For example, to specify to use the season period that contains the start date of the bill period, put the following ASSIGNMENT Statement in your schedule: BILL_PERIOD_SELECT = 1;
BILL_START	The first day (with time) of the account bill period.
BILL_STOP	The last day (with time) of the account bill period.
BILL_TYPE*	The type of process. Will always be CURRENT for bill determinant calculations.
CURRENT_DATE	The current system (i.e. today's) date.
EFFECTIVE_DATE	The last day in the bill period. Used to determine which factors and other effective-dated records to use in processing.
FULL_BILL_START	Midnight (00:00:00) on the first day of the account bill period.
FULL_BILL_STOP	Midnight (23:59:59) on the last day of the account bill period.
HOURS_PER_MONTH*	The default value is 730. To apply a different value, use an ASSIGNMENT Statement. Specifically, set HOURS_PER_MONTH equal to either a desired constant value, or to the results of the BILLINGHOURS or MONTHHOURS functions (see <b>BILLINGHOURS Function</b> on page 13-21 and <b>MONTHHOURS Function</b> on page 13-35, respectively, in the <i>Oracle Utilities Rules Language Reference Guide</i> ). For example, to specify the actual number of hours in the current billing period for the account, include the following ASSIGNMENT Statement in your schedule: HOURS_PER_MONTH = BILLINGHOURS();
NUMDAYS	The number of days in the bill period (BILL_STOP - BILL_START, rounded to the nearest day).
RATE_SCHEDULE_CODE	The code of the rate schedule being executed.



Identifier	Description
READ_DATE	The READDATE in the Bill History record. If Null, defaults to BILL_STOP.
RS_EFFECTIVE_START	Either the BILL_START or the rate schedule's effective start (as defined in the Rate Code History Table for the account), whichever is later.
RS_EFFECTIVE_STOP	Either the BILL_STOP or the rate schedule's effective stop (as defined in the Rate Code History Table for the account), whichever is earlier.
RS_JURIS_CODE	The rate schedule's jurisdiction code.
RS_OPCO_CODE	The rate schedule's operating company code.
UIDACCOUNT	The UID of the account being processed.

**Example:** The list below provide an example of the pre-defined identifiers available to bill determinant calculations.

```
ACCOUNT.ACCOUNTID = "MDM_ACCT_00" (STRING)
BILL_PERIOD = 05/01/2003 (DATE)
BILL_PERIOD_SELECT = 0 (INTEGER)
BILL_START = 05/01/2003 (DATE)
BILL_STOP = 05/31/2003 23:59:59 (DATE)
BILL_TYPE = "CURRENT" (STRING)
CURRENT_DATE = 06/05/2003 11:54:57 (DATE)
EFFECTIVE_DATE = 05/31/2003 (DATE)
FULL_BILL_START = 05/01/2003 (DATE)
FULL_BILL_STOP = 05/31/2003 23:59:59 (DATE)
GOVERNING_DATE = 05/31/2003 (DATE)
HOURS_PER_MONTH = 730.0 (FLOAT)
NUMDAYS = 31 (INTEGER)
RATE_SCHEDULE_CODE = "INTD_BD_CALC" (STRING)
READ_DATE = 05/31/2003 (DATE)
RS_EFFECTIVE_START = 05/01/2003 (DATE)
RS_EFFECTIVE_STOP = 05/31/2003 23:59:59 (DATE)
RS_JURIS_CODE = "ATLANTA" (STRING)
RS_OPCO_CODE = "METCO" (STRING)
UIDACCOUNT = 123 (INTEGER)
```

## Loading Account/Service Point/Meter

The first step in bill determinant calculations is to access the account, service point (if applicable), and meter for which the calculations are being performed.

### Loading the Account

The calculator engine executes the Rules Language in the content of the specified account, meaning that the account record can be accessed via database identifiers. For example, to assign the ACCT\_ID identifier to the value of the current account's ID, you could use the following:

```
//Set ACCT_ID
ACCT_ID = ACCOUNT.ACCOUNTID;
```

See **Database Identifiers** on page 4-5 in the *Oracle Utilities Rules Language User's Guide* for more information about database identifiers.

### Loading Service Points and Meters for an Account

Once the account ID has been assigned to an identifier, you can use the FOR EACH IN LIST statement or LISTVALUE function to access the service points and meters associated with the account. Use of these operations involves creating a Table-Column list or Query List that retrieves the service point or meter records in effect for the account during the bill period (defined by the

BILL\_START and BILL\_STOP pre-defined identifiers). See **Chapter 8: Working with Lists and Queries** in the *Data Manager User's Guide* for more information about creating Table-Column Lists. See **Query Lists** on page 7-66 in the *Oracle Utilities Energy Information Platform User's Guide* for more information creating query lists. See the **For Each x In List Statement** on page 3-10 in the *Oracle Utilities Rules Language Reference Guide* for more information about using the FOR EACH IN LIST statement, and the **LISTVALUE Function** on page 13-15 in the *Oracle Utilities Rules Language Reference Guide* for more information about using the LISTVALUE function.

The specific data you need to retrieve differs depending the type of record.

**Service Points:** When loading service points, you should assign identifiers to all the columns in the Service Point table that comprise the service point's identity, including:

- SERVICEPOINTID (Service Point ID)
- MARKETID (Market ID)
- SERVICETYPE (Service Type)

**Note:** Loading service points is only necessary if you need to calculate and save bill determinants at the service point level. Skip this if calculating and saving bill determinants for accounts and meters only.

**Example:** Loading Service Point records for an account:

```
//Load Service Point Records
//List: MDM_BD_SERVICEPOINT_FULL_FOR_ACCT_WITHIN_BP
//Criteria: Account ID, Start/Stop within Bill Period
//Returns active Service Point records for Account
FOR EACH SP_RECORD IN LIST
"MDM_BD_SERVICEPOINT_FULL_FOR_ACCT_WITHIN_BP"
  //Get Service Point ID, Market ID, and Service Type
  SP_ID = SP_RECORD.SERVICEPOINTID;
  MKT_ID = SP_RECORD.MARKETID;
  SERVICE_TYPE = SP_RECORD.SERVICETYPE;
  . . .
END FOR;
```

**Meters:** When loading meters, you should assign identifiers to all the columns in the MDM Meter table that comprise the meter's identity, including:

- METERID (Meter ID)
- EXPECTEDUOM (Expected UOM Code)
- CHANNELID (Channel ID)

**Example:** Loading Meter Data Channel records for an account:

```
//Load Meter Data Channel Records
//List: MDM_BD_MTRDATACHANNEL_FULL_FOR_ACCT_WITHIN_BP
//Criteria: Account ID, Start/Stop within Bill Period
//Returns active MDM Meter records for Account
FOR EACH MTR_RECORD IN LIST
"MDM_BD_MTRDATACHANNEL_FULL_FOR_ACCT_WITHIN_BP"
  //Get Meter Details and Report Them
  MET_ID = MTR_RECORD.METERID;
  CHAN_ID = MTR_RECORD.CHANNELID;
  UOM_CODE = MTR_RECORD.EXPECTEDUOM;
  . . .
END FOR;
```

## Loading Usage Data

Once you've retrieved the account, service point (if applicable), and meter, the next step is to load usage for the billing period. The specifics concerning how you load usage depend on the type of usage.

## Loading Interval Data

You use the enhanced interval data functions to load interval usage for a meter. These functions load interval data from the Meter Data Channel Cut table and use the identity of the meter as one of their parameters. You can create this identity by concatenating the Meter ID, Expected Unit of Measure, and Channel ID into a single string.

**Example:** Concatenate meter ID, UOM, and channel ID into a single string:

```
FOR EACH MTR_RECORD IN LIST
"MDM_BD_MTRDATACHANNEL_FULLL_FOR_ACCT_WITHIN_BP"
  //Get Meter Details
  MET_ID = MTR_RECORD.METERID;
  CHAN_ID = MTR_RECORD.CHANNELID;
  UOM_CODE = MTR_RECORD.EXPECTEDUOM;
  MTRDATACHAN = MET_ID + "," + UOM_CODE + "," + CHAN_ID;
  TABLENAME = "LSMDMTRDATAACUT";
  CATEGORY = "FINAL";
  . . .
END FOR;
```

The enhanced interval data functions also require that the table name ("LSMDMTRDATAACUT") and usage category ("FINAL") be specified when loading interval data. You can specify these using Assignment statements.

**Example:** Assigning identifiers for table name and usage category:

```
FOR EACH MTR_RECORD IN LIST
"MDM_BD_MTRDATACHANNEL_FULLL_FOR_ACCT_WITHIN_BP"
  //Get Meter Details
  MET_ID = MTR_RECORD.METERID;
  CHAN_ID = MTR_RECORD.CHANNELID;
  UOM_CODE = MTR_RECORD.EXPECTEDUOM;
  MTRDATACHAN = MET_ID + "," + UOM_CODE + "," + CHAN_ID;
  TABLENAME = "LSMDMTRDATAACUT";
  CATEGORY = "FINAL";
  . . .
END FOR;
```

With these identifiers set, you can use one of the enhanced interval data functions to load the data.

**Example:** Load interval data for an account (using INTDLOADEXDATES):

```
. . .
//Load interval data if UOM is KWH
IF UOM_CODE = "01"
  THEN
    THIS_MTR_KWH_HNDL = INTDLOADEXDATES(MTRDATACHAN , CATEGORY ,
TABLENAME , BILL_START , BILL_STOP);
    //
    . . .
  END IF;
```

See the **Enhanced Interval Data Functions** on page 9-85 in the *Oracle Utilities Rules Language Reference Guide* for more information about these functions.

## Loading Consumption and TOU Usage

For consumption and TOU usage, you need to retrieve both the meter readings for the billing period (from the Meter Data Readings table), as well as the meter configuration record that is in effect during the billing period (from the Meter Configuration table), since bill determinant calculations rely on values defined in a meter's configuration (such as Meter Multiplier or Number of Dials). For TOU usage, you also need to retrieve the individual TOU meter readings (from the Meter Data TOU Readings table). You can use the FOR EACH IN LIST statement or the LISTVALUE function to access these records. Use of these operations involves creating a Table-Column list or Query List that retrieves the meter read records that fall between the BILL\_START and BILL\_STOP pre-defined identifiers. See **Chapter 8: Working with Lists and Queries** in the *Data Manager User's Guide* for more information about creating Table-Column Lists. See **Query Lists** on page 7-66 in the *Oracle Utilities Energy Information Platform User's Guide* for more information creating query lists. See the **For Each x In List Statement** on page 3-10 in the *Oracle Utilities Rules Language Reference Guide* for more information about using the FOR EACH IN LIST statement, and the **LISTVALUE Function** on page 13-15 in the *Oracle Utilities Rules Language Reference Guide* for more information about using the LISTVALUE function.

The specific data you need to retrieve differs depending the type of record.

**Meter Configuration:** When loading meter configuration records, you should assign identifiers to all the columns in the Meter Configuration table required for your calculations, such as:

- METERMULTIPLIER (Meter Multiplier)
- NUMOFDIALS (Number of Dials)

**Example:** Loading Meter Configuration records for a meter:

```
FOR EACH MTR_RECORD IN LIST
"MDM_BD_MTRDATACHANNEL_FULL_FOR_ACCT_WITHIN_BP"
  //Get Meter Details
  MET_ID = MTR_RECORD.METERID;
  CHAN_ID = MTR_RECORD.CHANNELID;
  UOM_CODE = MTR_RECORD.EXPECTEDUOM;
  //
  //Load Meter Configuration Records
  //List: MDM_BD_MTRCFG_FULL_FOR_METERID
  //Criteria: Meter ID, Start/Stop within Bill Period
  //Returns active Meter Configuration records for Meter
  FOR EACH MTR_CFG IN LIST "MDM_BD_MTRCFG_FULL_FOR_METERID"
    MULT = MTR_CFG.METERMULTIPLIER;
    NUM_DIALS = MTR_CFG.NUMOFDIALS;
  END FOR;
  ...
END FOR;
```

**Meter Read:** When loading meter read records, you should assign identifiers to all the columns in the Meter Data Readings table required for your calculations, including:

- STARTVAL (Start Value)
- STOPVAL (Stop Value)

**Example:** Loading Meter Read records for a meter:

```
FOR EACH MTR_RECORD IN LIST
"MDM_BD_MTRDATACHANNEL_FULL_FOR_ACCT_WITHIN_BP"
  //Get Meter Details
  MET_ID = MTR_RECORD.METERID;
  CHAN_ID = MTR_RECORD.CHANNELID;
  UOM_CODE = MTR_RECORD.EXPECTEDUOM;
  ...
  //Load Meter Read Records
  //List: MDM_BD_MTRDATAREAD_FULL_FOR_METERID
```

```

//Criteria: Meter ID, Start/Stop within Bill Period
//Returns active Meter Read records within Bill Period
FOR EACH MET_READ IN LIST "MDM_BD_MTRDATAAREAD_FULL_FOR_METERID"
  //Get Meter Read Details and Report Them
  PREVVAL = MET_READ.STARTVAL;
  CURVAL = MET_READ.STOPVAL;
  ...
END FOR;
END FOR;

```

**TOU Meter Read:** When loading TOU meter read records, you should assign identifiers to all the columns in the Meter Data TOU Readings table required for your calculations, including:

- TOUPERIODNAME (TOU Period Name)
- STARTVAL (Start Value)
- STOPVAL (Stop Value)

**Example:** Loading TOU Meter Read records for a meter:

```

FOR EACH MTR_RECORD IN LIST
"MDM_BD_MTRDATACHANNEL_FULL_FOR_ACCT_WITHIN_BP"
  //Get Meter Details
  MET_ID = MTR_RECORD.METERID;
  CHAN_ID = MTR_RECORD.CHANNELID;
  UOM_CODE = MTR_RECORD.EXPECTEDUOM;
  ...
  //Load TOU Readings
  //List: MDM_BD_MTRDATATOUREAD_FULL_FOR_METERID
  //Criteria: Meter Reading
  //Returns active Meter TOU Read records for Meter Reading
  CATEGORY = "FINAL";
  FOR EACH TOU_READ IN LIST "MDM_BD_MTRDATATOUREAD_FULL_FOR_METERID"
    //Get Meter Read Details and Report Them
    TOU_PER_NAME = TOU_READ.TOUPERIODNAME;
    TOU_PREVVAL = TOU_READ.STARTVAL;
    TOU_CURVAL = TOU_READ.STOPVAL;
    ...
  END FOR;
END FOR;

```

## Calculating Billing Determinants

After the usage readings are loaded into memory, you can calculate the bill determinant values based on your business requirements. The specifics concerning how these calculations are performed depend on the type of usage.

## Calculating Bill Determinants for Interval Meters

After interval data has been loaded, you can perform operations on the data as dictated by your business requirements. Typical operations might include creating masks or performing block or scalar operations. See **Chapter 7: Working with Interval Data** in the *Oracle Utilities Rules Language User's Guide* for more information about working with interval data, and **Working with Enhanced/Generic Interval Data** on page 7-30 for specifics concerning enhanced interval data (which applies to Meter Data Management interval data).

After interval data has been loaded and operated on (if applicable), you can derive billing determinants and other values from the data using the following format:

```
<value> = HNDL.ATTRIBUTE;
```

where

- **<value>**: an identifier you assign to the value.
- **HNDL**: the interval data handle that you assigned when you loaded the data (KWH\_HNDL in the example below)
- **ATTRIBUTE**: the name of a particular attribute of the handle. See **Interval Data Reference Values and Attributes** on page 7-3 in the *Oracle Utilities Rules Language User's Guide* for a list of values that can be derived from interval data handles.

For example, one of the available values is TOTAL, which is the total of all the interval values in the handle. If you used the handle HNDL, you could retrieve the total value using an Assignment Statement as follows:

```
HNDL_TOTAL = HNDL.TOTAL;
```

See **Deriving Billing Determinants and Values from Interval Data** on page 7-36 in the *Oracle Utilities Rules Language User's Guide* for more information.

**Example:** Calculate the total KWH and demand (KW) for interval usage:

```
...
//Load interval data if UOM is KWH
IF UOM_CODE = "01"
THEN
    THIS_MTR_KWH_HNDL = INTDLOADEXDATES(MTRDATACHAN , CATEGORY ,
TABLENAME , BILL_START , BILL_STOP);
    //
    //Reassign to KWH_HNDL if not previously assigned
    IF HASVALUE(KWH_HNDL) = 0
    THEN
        KWH_HNDL = THIS_MTR_KWH_HNDL;
    ELSE
        KWH_HNDL = KWH_HNDL + THIS_MTR_KWH_HNDL;
    END IF;
    CLEAR THIS_MTR_KWH_HNDL;
END IF;
END FOR;
//
//Calculate Billing Determinants
RECORDED_KWH = KWH_HNDL.TOTAL;
//
//Calculate Demand
//Convert KWH_HNDL to KW_DMD_HNDL
KW_DMD_HNDL = KWH_HNDL * KWH_HNDL.IPH;
//
//Calculate Peak KW
PEAK_KW = KW_DMD_HNDL.MAX;
```

## Calculating Bill Determinants for Consumption and Time of Use Meters

The calculations for consumption and time of use bill determinants are based entirely on your business requirements. One of the most common calculations is calculating total consumption for a meter.

For consumption and TOU meters, you calculate the total consumption measured using the following formula:

$$\text{Total} = (\text{Stop Value} - \text{Start Value}) * \text{Meter Multiplier}$$

In the case of a meter roll-over (where the Stop Value is lower than the Start Value), you need to calculate an adjusted Stop Value by adding a value equal to 10 raised to the power of  $n$ , where  $n$  is the number of dials on the meter. This adjustment results in a Stop Value that is higher than the Start Value. For example, consider the following meter reading values for a meter with 5 dials:

- Start Value: 99900
- Stop Value: 00100

Applying the basic formula above would produce a negative result ( $00100 - 99900 = -99800$ ). Adjusting the Stop Value for this reading as described above would result in an adjusted Stop Value of 100100, which in turn can be used to properly calculate the total consumption for the meter reading ( $100100 - 99900 = 200$ ). You can perform this adjustment using the POW Rules Language function.

Any number of other calculations are also possible, including the use of constants (stored as Factors in the database) or other formulas, based on your business requirements.

The specific data used in your calculations differs depending the type of usage.

**Consumption:** For consumption meters, you would apply your calculations based on the STARTVAL and STOPVAL values from the meter read record.

**Example:** Calculate total consumption for a consumption meter:

```
//Load Meter Read Records
//List: MDM_BD_MTRDATAREAD_FULL_FOR_METERID
//Criteria: Meter ID, Start/Stop within Bill Period
//Returns active Meter Read records within Bill Period
FOR EACH MET_READ IN LIST "MDM_BD_MTRDATAREAD_FULL_FOR_METERID"
  //Get Meter Read Details and Report Them
  PREVVAL = MET_READ.STARTVAL;
  CURVAL = MET_READ.STOPVAL;
  //
  //Compute the consumption ((stop - start) * multiplier)
  IF CURVAL < PREVVAL
    THEN
      //Rollover has occurred. Adjust current value for calc
      purposes
      ADJ_CURVAL = CURVAL + POW(10 , NUM_DIALS);
    ELSE
      ADJ_CURVAL = CURVAL;
  END IF;
  THIS_MET_DIFF = ADJ_CURVAL - PREVVAL;
  THIS_MET_KWH = THIS_MET_DIFF * MULT;
  RECORDED_KWH = RECORDED_KWH + THIS_MET_KWH;
END FOR;
```

**Time of Use:** For time of use meters, you would apply your calculations based on the STARTVAL and STOPVAL values from the TOU meter read records.

**Example:** Calculate total consumption for a time of use meter:

```
//Load TOU Readings
//List: MDM_BD_MTRDATATOUREAD_FULL_FOR_METERID
//Criteria: Meter Reading
//Returns active Meter TOU Read records for Meter Reading
CATEGORY = "FINAL";
FOR EACH TOU_READ IN LIST "MDM_BD_MTRDATATOUREAD_FULL_FOR_METERID"
  //Get Meter Read Details and Report Them
  TOU_PER_NAME = TOU_READ.TOUPERIODNAME;
  TOU_PREVVAL = TOU_READ.STARTVAL;
  TOU_CURVAL = TOU_READ.STOPVAL;
  //
  //Compute the consumption ((stop - start) * multiplier)
  IF TOU_CURVAL < TOU_PREVVAL
    THEN
      //Rollover has occurred. Adjust current value for calc
purposes
      ADJ_TOU_CURVAL = TOU_CURVAL + POW(10 , NUM_DIALS);
    ELSE
      ADJ_TOU_CURVAL = TOU_CURVAL;
    END IF;
    THIS_MET_DIFF = ADJ_TOU_CURVAL - TOU_PREVVAL;
    THIS_MET_KWH = THIS_MET_DIFF * MULT;
    //
    //Calculate On Peak and Off Peak KWH
    SELECT TOU_PER_NAME
      WHEN "ON_PEAK"
        ON_PEAK_KWH = ON_PEAK_KWH + THIS_MET_KWH;
      WHEN "OFF_PEAK"
        OFF_PEAK_KWH = OFF_PEAK_KWH + THIS_MET_KWH;
    END SELECT;
  END FOR;
```

## Saving Billing Determinant Values

Once your bill determinant values have been calculated, the last step is to save the results to the database or to an export file.

### Saving Results to the Database

When saving results to the Oracle Utilities Data Repository, you can save at the account-level, service point-level, or the meter-level (or any combination of the three). The specific Rules Language functions used to save the results depend on the level at which you're saving the data.

**Account-Level:** Account-level results can be saved to the Bill History Values table using the SAVE or SAVE AS statements. See the **Save Statements** on page 6-3 in the *Oracle Utilities Rules Language Reference Guide* for more information about these statements.

**Example:** Save KWH (RECORDED\_KWH) and KW (RECORDED\_KW) for the account

```
//Calculate Billing Determinants
RECORDED_KWH = KWH_HNDL.TOTAL;
SAVE RECORDED_KWH;
//
//Calculate Demand
//Convert KWH_HNDL to KW_DMD_HNDL
KW_DMD_HNDL = KWH_HNDL * KWH_HNDL.IPH;
//
//Calculate Peak KW
PEAK_KW = KW_DMD_HNDL.MAX;
```



```

PEAK_KW_DT = KW_DMD_HNDL.MAXDATE;
//
RECORDED_KW = PEAK_KW;
SAVE RECORDED_KW;

```

**Service Point-Level:** Service point-level results can be saved to the Service Point Bill Determinant Values table using the SAVE TO TABLE statement. See the **Save Statements** on page 6-3 in the *Oracle Utilities Rules Language Reference Guide* for more information about this statement. When saving to the Service Point Bill Determinant Values table, you must provide values for the following columns:

- SERVICEPOINTID (Service Point ID)
- MARKETID (Market ID)
- SERVICETYPE (Service Type)
- ACCOUNTID (Account ID)
- STARTTIME (Start Time)
- BILLDETERMCODE (Bill Determinant Code)
- VALUE (Value)

**Example:** Save KWH (RECORDED\_KWH) for the service point

```

SP_BD.SERVICEPOINTID = SP_ID;
SP_BD.MARKETID = MKT_ID;
SP_BD.SERVICETYPE = SERVICE_TYPE;
SP_BD.ACCOUNTID = ACCT_ID;
SP_BD.STARTTIME = BILL_START;
SP_BD.BILLDETERMCODE = "RECORDED_KWH";
SP_BD.VALUE = RECORDED_KWH;
SAVE SP_BD TO TABLE "LSMDBDVALUESP";

```

**Meter-Level:** Meter-level results can be saved to the Meter Bill Determinant Values table using the SAVE TO TABLE statement. See the **Save Statements** on page 6-3 in the *Oracle Utilities Rules Language Reference Guide* for more information about this statement. When saving to the Meter Bill Determinant Values table, you must provide values for the following columns:

- METERID (Meter ID)
- EXPECTEDUOM (Expected UOM Code)
- CHANNELID (Channel ID)
- ACCOUNTID (Account ID)
- STARTTIME (Start Time)
- BILLDETERMCODE (Bill Determinant Code)
- VALUE (Value)

**Example:** Save KWH (RECORDED\_KWH) for the meter

```

MTR_BD.METERID = MET_ID;
MTR_BD.EXPECTEDUOM = UOM_CODE;
MTR_BD.CHANNELID = CHAN_ID;
MTR_BD.ACCOUNTID = ACCT_ID;
MTR_BD.STARTTIME = BILL_START;
MTR_BD.BILLDETERMCODE = "RECORDED_KWH";
MTR_BD.VALUE = RECORDED_KWH;
SAVE MTR_BD TO TABLE "LSMDBDVALUEMTR";

```

## Saving Results to files

You can also save the calculation results to files for use in other systems. You can save results in either a delimited or XML file.

**Note:** When saving results to a file, each Rules Language process (each account) should create its own uniquely named output file. Multiple concurrent processes writing to the same file can result in contention issues.

**Delimited File:** Saving results to a delimited (CIS) file involves creating a format file (that defines the format of the output file) and saving the data using the SAVE TO CIS statement. See the **Save Statements** on page 6-3 in the *Oracle Utilities Rules Language Reference Guide* for more information about this statement. See **Chapter 9: Creating a CIS Transaction Record Output File** in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about creating delimited output files.

**Note:** When **not** saving output to a delimited (CIS) file, the IGNORE\_CIS\_FILE parameter should be specified in the LODESTAR.CFG file. This allows bill determinant calculation processing to ignore any errors related to writing to the default output file. See **LODESTAR.CFG** on page 2-2 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information.

**XML File:** Saving results to an XML file involves defining the XML structure of the output and saving the data using the DOMDOCSAVEFILE function. See the **DOMDOCSAVEFILE Function** on page B-16 in the *Oracle Utilities Rules Language Reference Guide* for more information about this function. See **Appendix B: XML Statements and Functions** in the *Oracle Utilities Rules Language Reference Guide* for more information about working with XML in the Rules Language.

## Setting the Billed Flag

The Meter Data Reading (LSMDMTRDATAREAD) and Meter Data Channel Cut (LSMDMTRDATAACUT) tables have a Billed column that indicates if bill determinants have been calculated for the reading. This flag is used by the Bill Determinant Recalculation validation to determine if bill determinants should be recalculated if a replacement reading is received or if a Final reading is manually edited (see **Bill Determinant Recalculation** on page 3-24 in the *Oracle Utilities Meter Data Management User's Guide*). This flag is not automatically set, and must be manually set. This can be done as part of bill determinant calculations or as part of a separate process.

How you set this flag in Rules Language depends on the type of reading.

**Interval Readings:** For interval data readings, you can set the Billed flag using the INTDSETATTREX function.

**Example:** Set the Billed flag for the THIS\_MTR\_KWH\_HNDL interval data handle to "Y":

```

...
//Load interval data if UOM is KWH
IF UOM_CODE = "01"
  THEN
    THIS_MTR_KWH_HNDL = INTDLOAEXDATES(MTRDATACHAN , CATEGORY ,
TABLENAME , BILL_START , BILL_STOP);
    //
    SET_BILLED_FLAG = INTDSETATTREX (THIS_MTR_KWH_HNDL, BILLED, "Y");
    //
    ...
  END IF;

```

See the **INTDSETATTREX Function** on page 9-101 in the *Oracle Utilities Rules Language Reference Guide* for more information about using this function.

**Consumption and TOU Readings:** For consumption and TOU readings, you can set the Billed flag using the SAVE TO TABLE statement. When using this statement, you must specify all the identity columns in the Meter Data Reading table, including:

- FK\_MTRDATAR\_MTRDATCMETERID (Meter ID)
- FK\_MTRDATAR\_MTRDATCEXPECTEDUOM (Expected UOM Code)
- FK\_MTRDATAR\_MTRDATCCHANNELID (Channel ID)
- CATEGORYCODE (Usage Category)
- STOPREADTIME (Stop Read Time)

Note that this requires you retrieve the Usage Category (which should be "FINAL") and the Stop Read Time values for the reading.

**Example:** Set the Billed flag on a consumption reading to "Y":

```
BILLED_FLAG.FK_MTRDATAR_MTRDATCMETERID = MET_ID;
BILLED_FLAG.FK_MTRDATAR_MTRDATCEXPECTEDUOM = UOM_CODE;
BILLED_FLAG.FK_MTRDATAR_MTRDATCCHANNELID = CHAN_ID;
BILLED_FLAG.CATEGORYCODE = "FINAL";
BILLED_FLAG.STOPREADTIME = MTR_READ_TIME;
BILLED_FLAG.BILLED = "Y";
SAVE BILLED_FLAG TO TABLE "LSMDMTRDATAREAD";
```

See the **Save Statements** on page 6-3 in the *Oracle Utilities Rules Language Reference Guide* for more information about using the SAVE TO TABLE statement.

## Sample Rules Language Calculations

Below are some samples of bill determinant calculations.

### Interval Usage for an Account

The following example calculates KWH (RECORDED\_KWH) and KW (RECORDED\_KW) for a single interval meter account (with a single service point and one or more meters) and saves the results in the Bill History Values table.

```
//Oracle Utilities Meter Data Management
//Billing Determinant Calculations - Interval Data
//Calculates and saves RECORDED_KWH and RECORDED_KW
//Created by: Lou Prosperi
//Date: 02/26/2010
//
//Set ACCT_ID
ACCT_ID = ACCOUNT.ACCOUNTID;
//
//Load Meter Data Channel Records
//List: MDM_BD_MTRDATACHANNEL_FULL_FOR_ACCT_WITHIN_BP
//Criteria: Account ID, Start/Stop within Bill Period
//Returns active MDM Meter records for Account
FOR EACH MTR_RECORD IN LIST
"MDM_BD_MTRDATACHANNEL_FULL_FOR_ACCT_WITHIN_BP"
  //Get Meter Details
  MET_ID = MTR_RECORD.METERID;
  CHAN_ID = MTR_RECORD.CHANNELID;
  UOM_CODE = MTR_RECORD.EXPECTEDUOM;
  MTRDATACHAN = MET_ID + "," + UOM_CODE + "," + CHAN_ID;
  TABLENAME = "LSMDMTRDATAACUT";
  CATEGORY = "FINAL";
  //
  //Load interval data if UOM is KWH
  IF UOM_CODE = "01"
    THEN
```

```

        THIS_MTR_KWH_HNDL = INTDLOAEXDATES(MTRDATACHAN , CATEGORY ,
TABLENAME , BILL_START , BILL_STOP);
    //
    //
    SET_BILLED_FLAG = INTDSETATTREX (THIS_MTR_KWH_HNDL, BILLED, "Y");
    //
    //Reassign to KWH_HNDL if not previously assigned
    IF HASVALUE(KWH_HNDL) = 0
    THEN
        KWH_HNDL = THIS_MTR_KWH_HNDL;
    ELSE
        KWH_HNDL = KWH_HNDL + THIS_MTR_KWH_HNDL;
    END IF;
    CLEAR THIS_MTR_KWH_HNDL;
    END IF;
END FOR;
//
//Calculate Billing Determinants
RECORDED_KWH = KWH_HNDL.TOTAL;
SAVE RECORDED_KWH;
//
//Calculate Demand
//Convert KWH_HNDL to KW_DMD_HNDL
KW_DMD_HNDL = KWH_HNDL * KWH_HNDL.IPH;
//
//Calculate Peak KW
PEAK_KW = KW_DMD_HNDL.MAX;
PEAK_KW_DT = KW_DMD_HNDL.MAXDATE;
//
RECORDED_KW = PEAK_KW;
SAVE RECORDED_KW;

```

### Consumption Usage for an Account

The following example calculates KWH (RECORDED\_KWH) for a single consumption meter account (with a single service point and one or more meters) and saves the results in the Bill History Values table.

```

//Oracle Utilities Meter Data Management
//Billing Determinant Calculations - Consumption
//Created by: Lou Prospero
//Date: 02/26/2010
//
//Set ACCT_ID
ACCT_ID = ACCOUNT.ACCOUNTID;
//
//Load Meter Data Channel Records
//List: MDM_BD_MTRDATACHANNEL_FULL_FOR_ACCT_WITHIN_BP
//Criteria: Account ID, Start/Stop within Bill Period
//Returns active MDM Meter records for Account
FOR EACH MTR_RECORD IN LIST
"MDM_BD_MTRDATACHANNEL_FULL_FOR_ACCT_WITHIN_BP"
    //Get Meter Details
    MET_ID = MTR_RECORD.METERID;
    CHAN_ID = MTR_RECORD.CHANNELID;
    UOM_CODE = MTR_RECORD.EXPECTEDUOM;
    //
    //Load Meter Configuration Records
    //List: MDM_BD_MTRCFG_FULL_FOR_METERID
    //Criteria: Meter ID, Start/Stop within Bill Period
    //Returns active Meter Configuration records for Meter
    FOR EACH MTR_CFG IN LIST "MDM_BD_MTRCFG_FULL_FOR_METERID"
        MULT = MTR_CFG.METERMULTIPLIER;
    //

```

```

        NUM_DIALS = MTR_CFG.NUMOFDIALS;
    END FOR;
    //
    //Load Meter Read Records
    //List: MDM_BD_MTRDATAREAD_FULL_FOR_METERID
    //Criteria: Meter ID, Start/Stop within Bill Period
    //Returns active Meter Read records within Bill Period
    FOR EACH MET_READ IN LIST "MDM_BD_MTRDATAREAD_FULL_FOR_METERID"
        //Get Meter Read Details and Report Them
        PREVVAL = MET_READ.STARTVAL;
        CURVAL = MET_READ.STOPVAL;
        //
        //Compute the consumption ((stop - start) * multiplier)
        IF CURVAL < PREVVAL
            THEN
                //Rollover has occurred. Adjust current value for calc
purposes
                ADJ_CURVAL = CURVAL + POW(10 , NUM_DIALS);
            ELSE
                ADJ_CURVAL = CURVAL;
            END IF;
            THIS_MET_DIFF = ADJ_CURVAL - PREVVAL;
            THIS_MET_KWH = THIS_MET_DIFF * MULT;
            RECORDED_KWH = RECORDED_KWH + THIS_MET_KWH;
        END FOR;
    END FOR;
    //
    //Save Recorded KWH
    SAVE RECORDED_KWH;

```

### Time of Use Usage for an Account

The following example calculates KWH (RECORDED\_KWH), on peak KWH (ON\_PEAK\_KWH), and off peak KWH (OFF\_PEAK\_KWH) for a single time of use meter account (with a single service point and one or more meters) and saves the results in the Bill History Values table.

```

//Oracle Utilities Meter Data Management
//Billing Determinant Calculation - Time of User
//Calculates and saves RECORDED_KWH, ON_PEAK_KWH, and OFF_PEAK_KWH
//Created by: Lou Prosperi
//Date: 02/26/2010
//
//Set ACCT_ID
ACCT_ID = ACCOUNT.ACCOUNTID;
//
//Load Meter Data Channel Records
//List: MDM_BD_MTRDATACHANNEL_FULL_FOR_ACCT_WITHIN_BP
//Criteria: Account ID, Start/Stop within Bill Period
//Returns active MDM Meter records for Account
FOR EACH MTR_RECORD IN LIST
"MDM_BD_MTRDATACHANNEL_FULL_FOR_ACCT_WITHIN_BP"
    //Get Meter Details
    MET_ID = MTR_RECORD.METERID;
    CHAN_ID = MTR_RECORD.CHANNELID;
    UOM_CODE = MTR_RECORD.EXPECTEDUOM;
    //
    //Load Meter Configuration Records
    //List: MDM_BD_MTRCFG_FULL_FOR_METERID
    //Criteria: Meter ID, Start/Stop within Bill Period
    //Returns active Meter Configuration records for Meter
    FOR EACH MTR_CFG IN LIST "MDM_BD_MTRCFG_FULL_FOR_METERID"
        MULT = MTR_CFG.METERMULTIPLIER;

```

```

        REPORT MULT LABEL "Meter Multiplier";
        NUM_DIALS = MTR_CFG.NUMOFDIALS;
        REPORT NUM_DIALS LABEL "Number of Meter Dials";
    END FOR;
    //
    //Load Meter Read Records
    //List: MDM_BD_MTRDATAREAD_FULL_FOR_METERID
    //Criteria: Meter ID, Start/Stop within Bill Period
    //Returns active Meter Read records within Bill Period
    FOR EACH MET_READ IN LIST "MDM_BD_MTRDATAREAD_FULL_FOR_METERID"
        //Get Meter Read Details and Report Them
        PREVVAL = MET_READ.STARTVAL;
        CURVAL = MET_READ.STOPVAL;
        //
        //Compute the consumption ((stop - start) * multiplier)
        IF CURVAL < PREVVAL
            THEN
                //Rollover has occurred. Adjust current value for calc
purposes
                ADJ_CURVAL = CURVAL + POW(10 , NUM_DIALS);
            ELSE
                ADJ_CURVAL = CURVAL;
            END IF;
        THIS_MET_DIFF = ADJ_CURVAL - PREVVAL;
        THIS_MET_KWH = THIS_MET_DIFF * MULT;
        RECORDED_KWH = RECORDED_KWH + THIS_MET_KWH;
    END FOR;
    //
    //Load TOU Readings
    //List: MDM_BD_MTRDATATOUREAD_FULL_FOR_METERID
    //Criteria: Meter Reading
    //Returns active Meter TOU Read records for Meter Reading
    CATEGORY = "FINAL";
    FOR EACH TOU_READ IN LIST "MDM_BD_MTRDATATOUREAD_FULL_FOR_METERID"
        //Get Meter Read Details and Report Them
        TOU_PER_NAME = TOU_READ.TOUPERIODNAME;
        TOU_PREVVAL = TOU_READ.STARTVAL;
        TOU_CURVAL = TOU_READ.STOPVAL;
        //
        //Compute the consumption ((stop - start) * multiplier)
        IF TOU_CURVAL < TOU_PREVVAL
            THEN
                //Rollover has occurred. Adjust current value for calc
purposes
                ADJ_TOU_CURVAL = TOU_CURVAL + POW(10 , NUM_DIALS);
            ELSE
                ADJ_TOU_CURVAL = TOU_CURVAL;
            END IF;
        THIS_MET_DIFF = ADJ_TOU_CURVAL - TOU_PREVVAL;
        THIS_MET_KWH = THIS_MET_DIFF * MULT;
        //
        //Calculate On Peak and Off Peak KWH
        SELECT TOU_PER_NAME
            WHEN "ON_PEAK"
                ON_PEAK_KWH = ON_PEAK_KWH + THIS_MET_KWH;
            WHEN "OFF_PEAK"
                OFF_PEAK_KWH = OFF_PEAK_KWH + THIS_MET_KWH;
        END SELECT;
    END FOR;
    //
END FOR;
//

```

```
//Save Recorded KWH  
SAVE RECORDED_KWH;  
SAVE ON_PEAK_KWH;  
SAVE OFF_PEAK_KWH;
```

## Executing Bill Determinant Calculations using the AccountSelector Interface

The AccountSelector interface can be used to initiate bill determinant calculations via a scripting tool or in batch mode. This section describes the interface methods and arguments, and provides example scripts.

### Interface Methods

The methods, interface objects, and syntax for the AccountSelector interface are as follows:

#### Run

**Description:** Used to run bill determinant calculations for a single account, a list of accounts, or all accounts.

**Method Name:** Run

**Interface:** AccountSelector

**Assembly Name:** LSAccountSelector

**Program ID:** Lodestar.AccountSelector

**Syntax:**

```
HRESULT Run ([in] BSTR xmlDataSource,
             [in] BSTR BDRequest);
```

#### Run Recalculation

**Description:** Used to recalculate bill determinants for a specified account.

**Method Name:** RunRecalc

**Interface:** AccountSelector

**Assembly Name:** LSAccountSelector

**Program ID:** Lodestar.AccountSelector

**Syntax:**

```
HRESULT RunRecalc ([in] BSTR xmlDataSource,
                  [in] BSTR BDRecalc);
```

#### Run Calculation Usage

**Description:** Used to calculate bill determinants for a specific account based on billing dates specified in the request.

**Method Name:** RunCalcUsage

**Interface:** AccountSelector

**Assembly Name:** LSAccountSelector

**Program ID:** Lodestar.AccountSelector

**Syntax:**

```
HRESULT RunCalcUsage ([in] BSTR xmlDataSource,
                     [in] BSTR BDCalcReq);
```

### Interface Arguments

The functions/methods of the Oracle Utilities Rules Language and Analysis interfaces use the following arguments:



**xmlDataSource Argument**

The xmlDataSource argument is an xml string that contains database connection and other related information. A DTD, xml example, and data element descriptions for this argument can be found on page 15-7 in the *Oracle Utilities Energy Information Platform Configuration Guide*.

**BDRRequest**

The BDRRequest argument is an xml string that specifies the account(s) and read date to be used in bill determinant calculations.

**Example:**

```
<BDRRequest readdate='2008-01-26' senderId='Paul '><All /></BDRRequest>
```

**Data Elements and Attributes:**

Each of the data elements used in the BDRRequest argument is described below.

**BDRRequest:** Root element specifying the read date, optional sender ID, and accounts to process.

Attributes:

**readdate:** The Read Date for the calculation request. This must be in the following format: "YYYY-MM-DD."

**senderid:** An optional Id for the sender of the request

Elements:

**All:** Element that specifies that the request should process all accounts.

**List:** Element that specifies that the request should process a list of accounts.

Attributes:

**name:** The name of the account list to process

**Account:** Element that specifies that the request should process a specified single account.

Attributes:

**uid:** The unique ID (UID) of the account to process

**BDRRecalc**

The BDRRecalc argument is an xml string that specifies the service point id and billing period to be recalculated.

**Example:**

```
<BDRRecalc meterId='PP_MDM_D2,01,1' start='2008-01-01T00:00:00'  
stop='2008-01-31T23:59:59' />
```

**Data Elements and Attributes:**

Each of the data elements used in the BDRRecalc argument is described below.

**BDRRecalc:** Root element specifying the meter ID, start time and stop time for the recalculation.

Attributes:

**meterId:** The identity of meter (meter ID, UOM code, and channel ID) to recalculate.

**start:** The start time of the bill period to recalculate. This must be in the following format: "YYYY-MM-DDThh:mm:ss."

**stop:** The stop time of the bill period to recalculate. This must be in the following format: "YYYY-MM-DDThh:mm:ss."

## BDCalcReq

The BDCalcReq argument is an xml string that specifies the account and billing dates to be used in bill determinant calculations.

### Example:

```
<request>
  <usageId>12345</usageId>
  <saId>12345</saId>
  <startDateTime>2008-01-01T00:00:00</startDateTime>
  <endDateTime>2008-01-31T23:59:59</endDateTime>
  <dateBreaks>
    </breakDateTime>
  </dateBreaks>
  <externalReferenceId>54321</externalReferenceId>
  <createDateTime>2008-02-03T00:00:00</createDateTime>
  </customElements>
</request>
```

### Data Elements and Attributes:

Each of the data elements used in the BDRcalc argument is described below.

**request:** Root element specifying the service point ID, start time and stop time for the recalculation.

Elements:

**usageId:** The id of the request.

**saId:** The account id.

**startDateTime:** The start time of the bill period to calculate. This must be in the following format: "YYYY-MM-DDThh:mm:ss."

**stopDateTime:** The stop time of the bill period to calculate. This must be in the following format: "YYYY-MM-DDThh:mm:ss."

**dateBreaks:** Element contain zero or more rate breaks between the startDateTime and endDateTime.

Elements:

**breakDateTime:** The date and time on which a rate change occurred for the account. This must be in the following format: "YYYY-MM-DDThh:mm:ss."

**externalReferenceID:** The batch ID of the process that submitted the request.

**createDateTime:** The date and time the request was created. This must be in the following format: "YYYY-MM-DDThh:mm:ss."

**customElements:** Element containing zero or more custom elements.

## Examples

This section contains examples of scripts that invoke the AccountSelector interface.

### Initiate Calculations for All accounts

The following examples initiate bill determinant calculations for all accounts for a read date of January 26, 2008.

#### Microsoft PowerShell:

```
$Selector = New-Object -ComObject "Lodestar.AccountSelector"

$xmlDataSource = "<DATASOURCE>" +
"<NAME>ORCL</NAME>" +
"<CONNECTSTRING>Data Source=PPP66;User
Id=LS106b1;Password=lodestar;LSPProvider=ODP;</CONNECTSTRING>" +
"<QUALIFIER>LS106b1</QUALIFIER>" +
"</DATASOURCE>"

$request = "<BDRequest readdate='2008-01-26' senderId='Paul'><All /></
BDRequest>";
$ret = $Selector.Run($xmlDataSource,$request)
```

#### JavaScript:

```
var selector = new ActiveXObject("Lodestar.AccountSelector");

var xmlDataSource =
"<DATASOURCE>" +
"<NAME>ORCL</NAME>" +
"<CONNECTSTRING>Data Source=PP66;User
Id=LS106B1;Password=lodestar;LSPProvider=ODP;</CONNECTSTRING>" +
"<QUALIFIER>LS106B1</QUALIFIER>" +
"</DATASOURCE>";

var request = "<BDRequest readdate='2008-01-26' senderId='Paul'><All
/></BDRequest>";
ret = selector.Run(xmlDataSource, request);
```

### Initiate Calculations for List of accounts

The following examples initiate bill determinant calculations for a list of accounts (THREE\_ACCOUNTS) for a read date of January 26, 2008.

#### Microsoft PowerShell:

```
$Selector = New-Object -ComObject "Lodestar.AccountSelector"

$xmlDataSource = "<DATASOURCE>" +
"<NAME>ORCL</NAME>" +
"<CONNECTSTRING>Data Source=PPP66;User
Id=LS106b1;Password=lodestar;LSPProvider=ODP;</CONNECTSTRING>" +
"<QUALIFIER>LS106b1</QUALIFIER>" +
"</DATASOURCE>"

$request = "<BDRequest readdate='2008-01-26' senderId='Paul'><List
name = 'THREE_ACCOUNTS' /></BDRequest>";
$ret = $Selector.Run($xmlDataSource,$request)
```

**JavaScript:**

```
var selector = new ActiveXObject("Lodestar.AccountSelector");

var xmlDataSource =
"<DATASOURCE>" +
" <NAME>ORCL</NAME>" +
" <CONNECTSTRING>Data Source=PP66;User
Id=LS106B1;Password=lodestar;LSProvider=ODP;</CONNECTSTRING>" +
" <QUALIFIER>LS106B1</QUALIFIER>" +
"</DATASOURCE>";

var request = "<BDRequest readdate='2008-01-27' senderId='Paul'><List
name='THREE_ACCOUNTS' /></BDRequest>";
ret = selector.Run(xmlDataSource, request);
```

**Initiate Calculations for a single account**

The following examples initiate bill determinant calculations for a single account (UIDACCOUNT=1) for a read date of January 26, 2008.

**Microsoft PowerShell:**

```
$Selector = New-Object -ComObject "Lodestar.AccountSelector"

$xmlDataSource = "<DATASOURCE>" +
"<NAME>ORCL</NAME>" +
"<CONNECTSTRING>Data Source=PPP66;User
Id=LS106b1;Password=lodestar;LSProvider=ODP;</CONNECTSTRING>" +
"<QUALIFIER>LS106b1</QUALIFIER>" +
"</DATASOURCE>"

$request = "<BDRequest readdate='2008-01-26' senderId='Paul'><Account
uid = '1' /></BDRequest>";
$ret = $Selector.Run($xmlDataSource,$request)
```

**JavaScript:**

```
var selector = new ActiveXObject("Lodestar.AccountSelector");

var xmlDataSource =
"<DATASOURCE>" +
" <NAME>ORCL</NAME>" +
" <CONNECTSTRING>Data Source=PP66;User
Id=LS106B1;Password=lodestar;LSProvider=ODP;</CONNECTSTRING>" +
" <QUALIFIER>LS106B1</QUALIFIER>" +
"</DATASOURCE>";

var request = "<BDRequest readdate='2008-01-26 00:00:00'
senderId='Paul'><Account uid='1' /></BDRequest>";
ret = selector.Run(xmlDataSource, request);
```

## Initiate Recalculation for a Meter

The following examples initiate bill determinant recalculations for a meter (PP\_MDM\_D2,01,1) for the billing period of January 2008 (1/1/2008 00:00:00 through 1/31/2008 23:59:59).

### Microsoft PowerShell:

```
$Selector = New-Object -ComObject "Lodestar.AccountSelector"

$xmlDataSource = "<DATASOURCE>" +
"<NAME>ORCL</NAME>" +
"<CONNECTSTRING>Data Source=PPP66;User
Id=LS106b1;Password=lodestar;LSPProvider=ODP;</CONNECTSTRING>" +
"<QUALIFIER>LS106b1</QUALIFIER>" +
"</DATASOURCE>"

$request = "<BDRecalc meterId='PP_MDM_D2,01,1' start='2008-01-
01T00:00:00' stop='2008-01-31T23:59:59' />"
$ret = $Selector.RunRecalc($xmlDataSource,$request)
```

### JavaScript:

```
var selector = new ActiveXObject("Lodestar.AccountSelector");

var xmlDataSource =
"<DATASOURCE>" +
"<NAME>ORCL</NAME>" +
"<CONNECTSTRING>Data Source=PP66;User
Id=LS106B1;Password=lodestar;LSPProvider=ODP;</CONNECTSTRING>" +
"<QUALIFIER>LS106B1</QUALIFIER>" +
"</DATASOURCE>";

var request = "<BDRecalc meterId='PP_MDM_D2,01,1' start='2008-01-
01T00:00:00' stop='2008-01-31T23:59:59' />";
var ret = selector.RunRecalc(xmlDataSource, request);
```

## Executing Bill Determinant Generation using the BDWS Web Service

The BDWS web service can be used to initiate bill determinant calculations for accounts. This web service inserts a record in the Payload Extension table. The BD\_SELECTOR\_TRIGGER service monitors this table, and when a record is detected, the BD\_SELECTOR\_TRIGGER business rule triggers the account selector to create a record in the BD Request table and records in the BD Queue / BD Queue Archive tables for each account. See the **BD\_SELECTOR\_TRIGGER** on page 6-12 for more information about this Adapter service.

This section outlines the technical details and use of the BDWS web service.

### Technical Details

This section outlines the technical details of the BDSW web service, including:

- **Web Service:** BDWS
- **Target Namespace:** <http://www.lodestarcorp.com/>
- **Port:** BDWSSoap
- **Location:** [http://\[MACHINE\\_NAME\]/lodestar/Products/BD/WebServices/bdws.asmx](http://[MACHINE_NAME]/lodestar/Products/BD/WebServices/bdws.asmx)  
where:
  - **[MACHINE\_NAME]** is the name (or IP address) of the machine on which the web service will be executed
- **Protocol:** SOAP
- **Transport protocol:** SOAP over HTTP

### SOAP Envelope

All BDSW web service requests must be sent within a SOAP envelope. Below is an example of the SOAP Envelope and Header information that must be contained included in the web service request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
  <soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd" xmlns="http://docs.oasis-open.org/wss/
2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:env="http://
schemas.xmlsoap.org/soap/envelope/" soap:mustUnderstand="1">
      <wsse:UsernameToken xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns="http://docs.oasis-open.org/
wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
        <wsse:Username>[USER_ID]</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-username-token-profile-1.0#PasswordText">[PASSWORD]</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    [REQUEST]
  </soap:Body>
</soap:Envelope>
```

where:

- **[USER\_ID]** is the web user ID of the user submitting the request
- **[PASSWORD]** is the user's password
- **[REQUEST]** is the web service request, based on the specific operation being called.

In addition, responses to web service requests are contained with a SOAP envelope. Below is an example of the SOAP Envelope and Header information sent with each response.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/
addressing" xmlns:wssse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/
01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <env:Header xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    <wsa:Action>http://www.lodestarcorp.com/BDWSSoap/
BDCalcAccountAllRequestResponse</wsa:Action>
    <wsa:MessageID>urn:uuid:a2315f6b-2b88-4cf5-9a44-c390af252295</
wsa:MessageID>
    <wsa:RelatesTo>uuid:0def747f-d25c-4841-8b58-3ea6c522065e</wsa:RelatesTo>
    <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</
wsa:To>
    <wsse:Security>
      <wsu:Timestamp wsu:Id="Timestamp-0f54874e-7915-4e48-be2b-d6945b3b9075">
        <wsu:Created>2010-03-17T19:33:22Z</wsu:Created>
        <wsu:Expires>2010-03-17T19:38:22Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </env:Header>
  <soap:Body>
    [RESPONSE]
  </soap:Body>
</soap:Envelope>
```

where:

- **[RESPONSE]** is the web service response, based on the specific operation being called.

## Operations

The BDSW web service provides the following operations:

- **BDCalcAccount**
- **BDCalcAccountAll**
- **BDCalcAccountList**

This section describes each operation and the parameters and format of the web request used for each.

### BDCalcAccount

Used to initiate bill determinant calculations for a single account.

#### Input Parameters:

The BDCalcAccount operation uses the following parameters:

Parameter	Type	Description
uidAccount	Input (required)	Unique ID (UID) of the account to be processed
readDate	Input (required)	The read date for the request. Must be in the following format: YYY-MM-DDThh:mm:ss

**Request XML Example:**

```
<soap:Envelope ...>
  <soap:Header ...>
  <soap:Body>
    <BDCalcAccount xmlns="http://www.lodestarcorp.com/">
      <uidAccount>1</uidAccount>
      <readDate>2008-01-26T00:00:00</readDate>
    </BDCalcAccount>
  </soap:Body>
</soap:Header>
</soap:Envelope>
```

**Output Parameters:**

The BDCalcAccount operation returns the following values:

Parameter	Description
BDCalcAccountResult	The UID of the Payload Extension record created by the request.

**Response XML Example:**

```
<soap:Envelope ...>
  <soap:Header ...>
  <soap:Body>
    <BDCalcAccountResponse xmlns="http://www.lodestarcorp.com/">
      <BDCalcAccountResult>3881</BDCalcAccountResult>
    </BDCalcAccountResponse>
  </soap:Body>
</soap:Header>
</soap:Envelope>
```

**BDCalcAccountAll**

Used to initiate bill determinant calculations for all accounts.

**Parameters:**

The BDCalcAccountAll operation uses the following parameters:

Parameter	Type	Description
readDate	Input (required)	The read date for the request. Must be in the following format: YYY-MM-DDThh:mm:ss

**Request XML Example:**

```
<soap:Envelope ...>
  <soap:Header ...>
  <soap:Body>
    <BDCalcAccountAll xmlns="http://www.lodestarcorp.com/">
      <readDate>2008-01-26T00:00:00</readDate>
    </BDCalcAccountAll>
  </soap:Body>
</soap:Header>
</soap:Envelope>
```



**Output Parameters:**

The BDCalcAccountAll operation returns the following values:

Parameter	Description
BDCalcAccountAllResult	The UID of the Payload Extension record created by the request.

**Response XML Example:**

```
<soap:Envelope ...>
  <soap:Header ...>
  <soap:Body>
    <BDCalcAccountAllResponse xmlns="http://www.lodestarcorp.com/">
      <BDCalcAccountAllResult>3901</BDCalcAccountAllResult>
    </BDCalcAccountAllResponse>
  </soap:Body>
</soap:Header>
</soap:Envelope>
```

**BDCalcAccountList**

Used to initiate bill determinant calculations for a list of accounts.

**Parameters:**

The BDCalcAccountList operation uses the following parameters:

Parameter	Type	Description
accountList	Input (required)	String containing the name of the list of accounts to process
readDate	Input (required)	The read date for the request. Must be in the following format: YYY-MM-DDThh:mm:ss

**Request XML Example:**

```
<soap:Envelope ...>
  <soap:Header ...>
  <soap:Body>
    <BDCalcAccountList xmlns="http://www.lodestarcorp.com/">
      <accountList>THREE_ACCOUNTS</accountList>
      <readDate>2008-01-26T00:00:00</readDate>
    </BDCalcAccountList>
  </soap:Body>
</soap:Header>
</soap:Envelope>
```

**Output Parameters:**

The BDCalcAccountList operation returns the following values:

Parameter	Description
BDCalcAccountListResult	The UID of the Payload Extension record created by the request.

**Response XML Example:**

```
<soap:Envelope ...>
  <soap:Header ...>
  <soap:Body>
    <BDCalcAccountListResponse xmlns="http://www.lodestarcorp.com/">
      <BDCalcAccountListResult>3882</BDCalcAccountListResult>
    </BDCalcAccountListResponse>
  </soap:Body>
</soap:Header>
</soap:Envelope>
```

# Chapter 7

---

## Configuring Oracle Utilities Meter Data Management Security

This chapter describes how to configure security to work with Oracle Utilities Meter Data Management, including:

- **Oracle Utilities Meter Data Management Security**

# Oracle Utilities Meter Data Management Security

This section describes the securable features of the Oracle Utilities Meter Data Management application, including:

- **Meter Data Management Features**
- **Important Notes about Assigning Meter Data Management Permissions**

## Meter Data Management Features

Oracle Utilities Meter Data Management features include:

- **Meter Management:** Enables the Meter Management menu, and expands to reveal Meter Management menu permissions including:
  - **Meters:** Enables the Meters menu item.
  - **Meter Configuration:** Enables the Meter Configuration menu item.
  - **Validation Groups:** Enables the Validation Groups menu item.
- **Usage Management:** Enables the Usage Management menu, and expands to reveal Usage Management menu permissions including:
  - **Save:** Enables the Save function on the usage management screens.
  - **Delete:** Enables the Delete function on the usage management screens.
  - **Approve:** Enables the Approve function on the usage management screens.
  - **Validate:** Enables the Validate function on the usage management screens.
  - **Usage Add/Edit:** Enables the Usage Add/Edit menu item.
  - **Usage Analysis:** Enables the Usage Analysis menu item.
    - **Graph:** Enables the Analysis Graph function on the usage management screens.
    - **Export:** Enables the Export to CSV function on the usage management screens.
    - **Usage Analysis Results:** Enables the Usage Analysis results screen.
      - **Edit Usage:** Enables the Edit Usage screens.
- **Operations:** Enables the Operations menu, and expands to reveal Operations menu permissions including:
  - **Usage Exceptions:** Enables the Usage Exceptions menu item.
  - **Physical Meter Events:** Enables the Physical Meter Events menu item.
  - **Process Control Interface:** Enables the Process Control Interface menu item.
  - **Work Queue:** Enables the MDM Work Queue menu item.
  - **My Work Queue Items:** Enables the My WQ Items menu item.
- **Revenue Protection Events:** Enables the Revenue Protection Events menu, and expands to reveal Revenue Protection Events menu permissions including:
  - **Search Events:** Enables the Search Events menu item.
- **Assets:** Enables the Meter Management menu, and expands to reveal Meter Management menu permissions including:
  - **Physical Meters:** Enables the Physical Meters menu item.
  - **Physical Meter Location History:** Enables the Physical Meter Location History menu item.

- **Meter Read Systems:** Enables the Meter Read Systems menu item.
- **Meter Manufacturers:** Enables the Meter Manufacturers menu item.
- **Setup:** Enables the Setup menu, and expands to reveal Setup menu permissions including:
  - **Meter Exception Codes:** Enables the Meter Exceptions menu item.
  - **Meter Events:** Enables the Meter Events menu item.
  - **Meter Event Types:** Enables the Meter Event Types menu item.
  - **Physical Meter Types:** Enables the Physical Meter Types menu item.
  - **Physical Meter Locations:** Enables the Physical Meter Locations menu item.
  - **Physical Meter Location Types:** Enables the Physical Meter Location Types menu item.
  - **Physical Meter Statuses:** Enables the Physical Meter Statuses menu item.
  - **Read Cycle Frequencies:** Enables the Read Cycle Frequencies menu item.
  - **Read Cycles:** Enables the Read Cycles menu item.
  - **Read Cycle Dates:** Enables the Read Cycle Dates menu item.
- **Billing Determinants:** Enables the Billing Determinants menu, and expands to reveal Billing Determinants menu permissions including:
  - **Initiation:** Enables the Initiate menu item
    - **All:** Enables the “All” option in the Account drop-down list on Initiate Billing Determinant Generation screen.
  - **Requests:** Enables the Requests menu item
  - **Results and Values:** Enables the Results and Values menu item
  - **Scheduled Jobs:** Enables the Scheduled Jobs menu item
- **Run Reports:** Enables the Run Reports option on the Meter Data Management menu.
- **View Reports:** Enables the View Reports option on the Meter Data Management menu.

## Important Notes about Assigning Meter Data Management Permissions

By default, the Meter Data Management features restrict access to all Oracle Utilities Meter Data Management functions and screens. To allow users access to Meter Data Management functionality, create users and groups and enable appropriate permissions for each.



# Appendix A

---

## Oracle Utilities Data Repository Meter Data Management Database Schema

This appendix includes a diagram of the Oracle Utilities Data Repository Meter Data Management database schema (v1.6.1.0.0) that provides details regarding the table and columns in the Oracle Utilities Meter Data Management schema, as well as the relationships between these tables in the Oracle Utilities Data Repository. This information is very useful when writing Rules Language statements or constructing database queries. This includes:

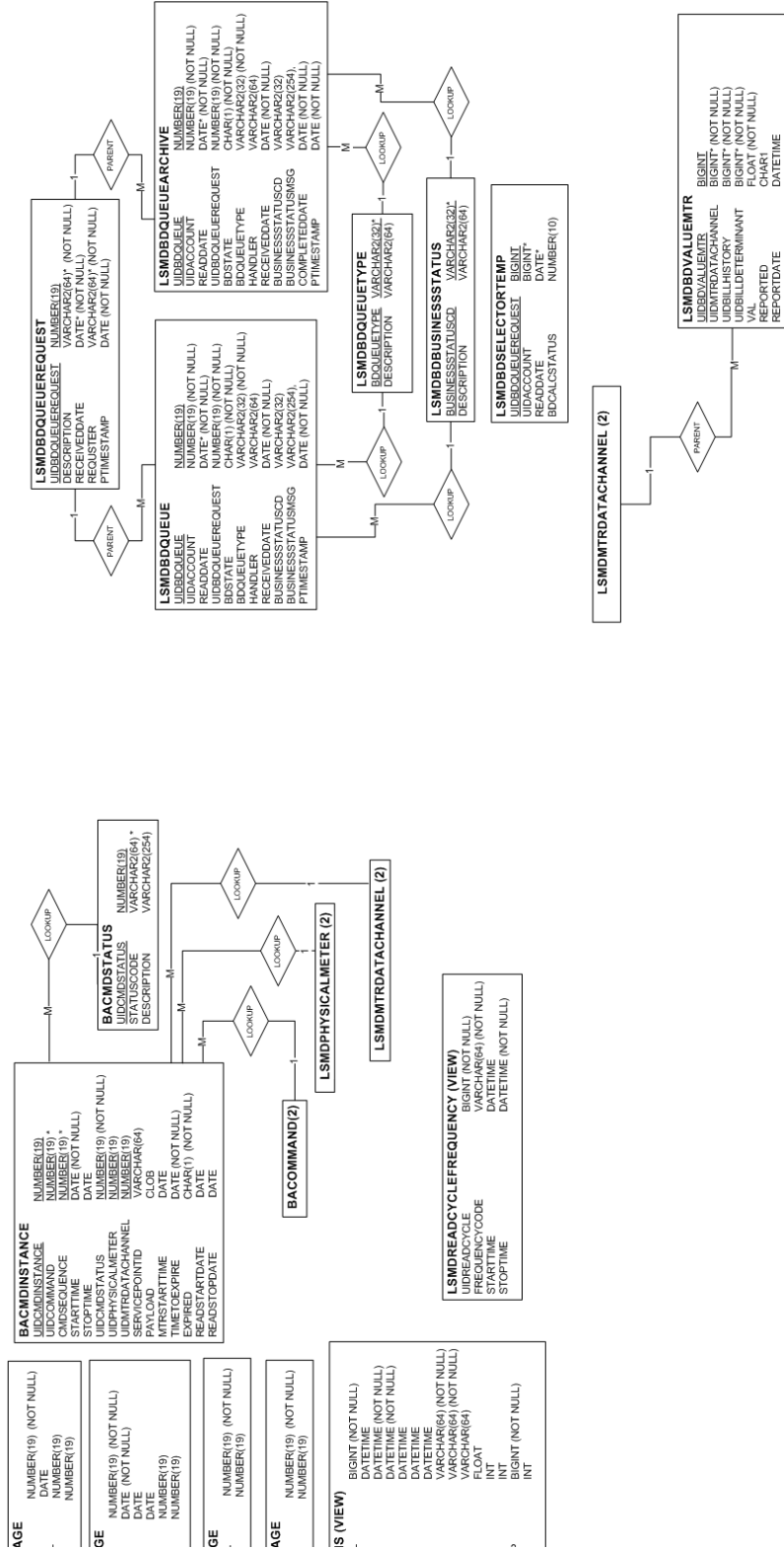
- **Oracle Utilities Meter Data Management Database Schema, page 1**
- **Oracle Utilities Meter Data Management Database Schema, page 2**





# Oracle Utilities Meter Data Management Database Schema, page 2

Utilities Meter Data Management v1.6.0.0.0  
 of 2)  
 Utilities Proprietary & Confidential





# Appendix B

---

## Integrating Oracle Utilities Meter Data Manager with a Customer Information System

This appendix describes the integration between Oracle Utilities Meter Data Management and Oracle Utilities Customer Care and Billing, including:

- **Overview**
- **Synchronizing Account and Service Point Data**
- **Processing Bill Determinant Requests**
- **Processing Replacement Readings**
- **Accessing Oracle Utilities Meter Data Management from External Applications**

**Note:** While this appendix specifically references Oracle Utilities Customer Care and Billing, the integration points described can also support integration with other customer information systems (CIS).

## Overview

In an integration between Oracle Utilities Meter Data Management and Oracle Utilities Customer Care and Billing:

- Oracle Utilities Meter Data Management is typically the “system of record” for meter-related data, including meter records, meter configurations, validation, editing, and estimation (VEE) rules, bill determinant calculation rules, usage data, and calculated bill determinants.
- Oracle Utilities Customer Care and Billing is typically the “system of record” for account-related and service point-related data, including the rates and tariffs used to calculate bills for each account and customer.

Given this breakdown of data between the two systems, any integration between them must account for the passage of data between the two to ensure that each system can accurately perform its business functions. The integration between Oracle Utilities Meter Data Management and Oracle Utilities Customer Care and Billing is based on four core business processes:

- **Synchronizing Account and Service Point Data**
- **Processing Bill Determinant Calculation Requests**
- **Processing Replacement Readings**
- **Accessing Oracle Utilities Meter Data Management from External Applications**

### Synchronizing Account and Service Point Data

In most integrations with Oracle Utilities Customer Care and Billing (or other CIS), Oracle Utilities Meter Data Management is not used as the system of record for account-related and service point-related data. Synchronizing account and service point data between the two systems ensures that all account-related and service point-related data in Oracle Utilities Meter Data Management is correct and up to date before bill determinant calculations are performed. This synchronization process is supported through a set of pre-configured Adapter services and business rules. See **Synchronizing Account and Service Point Data** on page B-4 for more information about setting up and enabling these services.

### Processing Bill Determinant Calculation Requests

Oracle Utilities Customer Care and Billing uses bill determinant data (usage totals based on meter readings) calculated and stored in Oracle Utilities Meter Data Management when calculating bills for customers. This process allows Oracle Utilities Customer Care and Billing to send requests for bill determinant calculations to Oracle Utilities Meter Data Management, which in turn performs the requests calculations, and publishes the results back to Oracle Utilities Customer Care and Billing. Processing bill determinant requests is supported through a set of pre-configured Adapter services and business rules. See **Processing Bill Determinant Requests** on page B-20 for more information about setting up and enabling these services.

### Processing Replacement Readings

In the event that replacement readings for an account are received by Oracle Utilities Meter Data Management, it may be necessary to recalculate previously calculated bill determinants for that account. This process provides a mechanism for Oracle Utilities Meter Data Management to alert Oracle Utilities Customer Care and Billing upon the receipt of replacement readings. Processing replacement readings is supported by a set of pre-configured Adapter services and business rules. See **Processing Replacement Readings** on page B-32 for more information about setting up and enabling these services.

---

## Accessing Oracle Utilities Meter Data Management from External Applications

From time to time, Oracle Utilities Customer Care and Billing users may need to view usage data in Oracle Utilities Meter Data Management. Accessing Oracle Utilities Meter Data Management directly from Oracle Utilities Customer Care and Billing is supported through configuration of a dashboard zone in Oracle Utilities Customer Care and Billing. See **Accessing Oracle Utilities Meter Data Management from External Applications** on page B-36 for more information.

## Synchronizing Account and Service Point Data

In most integrations with Oracle Utilities Customer Care and Billing (or other CIS), Oracle Utilities Meter Data Management is not used as the system of record for account-related and service point-related data. Synchronizing account and service point data between the two systems ensures that all account-related and service point-related data in Oracle Utilities Meter Data Management is correct and up to date before bill determinant calculations are performed. This section describes how data synchronization between Oracle Utilities Meter Data Management and Oracle Utilities Customer Care and Billing is performed, including:

- **Synchronization Adapter Services and Business Rules**
- **Synchronization Payload Formats**
- **Configuring Synchronization Adapter Services**
- **Synchronization Adapter RDL Schemas**

### Synchronization Adapter Services and Business Rules

Oracle Utilities Meter Data Management uses Adapter services and business rules to handle data synchronization requests (sync requests) from Oracle Utilities Customer Care and Billing. The descriptions of these services and rules include default settings for Runtime Service properties and Business Rule properties. See **Defining Service Properties for Adapter Runtime Services** on page 11-28 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about runtime service properties, and **Defining Properties and Parameters for Adapter Business Rules** on page 11-107 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about business rule properties.

**Note:** Multiple copies of each service can be active at the same time, to spread account processing across servers and services. See **Configuring Synchronization Adapter Services** on page B-12 for more information.

#### CCB\_SP\_TO\_MDM\_SERVICE

When service point sync requests are sent from Oracle Utilities Customer Care and Billing, they are sent to a JMS queue monitored by the CCB\_SP\_TO\_MDM\_SERVICE service. When a record is detected in the queue, the CCB\_SP\_TO\_MDM\_SERVICE business rule updates the service point-related data based on the details in the sync request. The CCB\_SP\_TO\_MDM\_SERVICE uses the following Runtime Service, Business Rule, and Rule Description Language.

#### CCB\_SP\_TO\_MDM\_SERVICE - Runtime Service

The CCB\_SP\_TO\_MDM Runtime Service is defined as follows:

- **Name:** CCB\_SP\_TO\_MDM\_SERVICE
- **JMS Queues:** <QUEUE\_NAME>
- **JMS Filter:** N/A
- **JMS Parameters:** N/A
- **Origination System:** ADAPTER
- **Parameters:** N/A
- **Internal Flag:** Y
- **Output Flag:** Y
- **Runtime Service Type:** QPORTAL
- **Properties:**  
<PROPERTIES>

```

<PROPERTY NAME="STORAGE_LEVEL" VALUE="2" />
<PROPERTY NAME="RULE_NAME" VALUE="CCB_SP_TO_MDM" />
<PROPERTY NAME="INPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.XSLConverter,DB://CCB_SA_SP_SYNC_PRE" />
<PROPERTY NAME="MAX_ERRORS" VALUE="50" />
<PROPERTY NAME="POLL_INTERVAL" VALUE="15" />
<PROPERTY NAME="OUTPUT_MECHANISM" VALUE="Q" />
<PROPERTY NAME="OUTPUT_QUEUE" VALUE="[JMS Server Name]/[JMS Module
Name]![Queue Name]" />
<PROPERTY NAME="OUTPUT_QUEUE_MSGTYPE" VALUE="T" />
<PROPERTY NAME="OUTPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.XSLConverter,DB://
CCB_SA_SP_SYNC_SUCCESS" />
<PROPERTY NAME="EXCEPTION_OUTPUT_MECHANISM" VALUE="Q" />
<PROPERTY NAME="EXCEPTION_OUTPUT_QUEUE" VALUE="[JMS Server Name]/[JMS
Module Name]![Queue Name]" />
<PROPERTY NAME="EXCEPTION_OUTPUT_QUEUE_MSGTYPE" VALUE="T" />
<PROPERTY NAME="EXCEPTION_OUTPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.XSLConverter,DB://
CCB_SA_SP_SYNC_ERROR" />
<PROPERTY NAME="EXCEPTION_OUTPUT_DC_1"
VALUE="com.lodestarcorp.portal.data.XSLConverter,DB://
CCB_SA_SP_SYNC_ERROR2" />
</PROPERTIES>

```

### CCB\_SP\_TO\_MDM - Business Rule

The CCB\_SP\_TO\_MDM Business Rule is defined as follows:

- **Rule Name:** CCB\_SP\_TO\_MDM
- **Description:** Synchronizes CC&B Service Point data into MDM data repository.
- **Rule Type:** Import
- **Properties:**

```

<PROPERTIES>
  <PROPERTY NAME="xmlstream" VALUE="n" />
  <PROPERTY NAME="mode" VALUE="event" />
</PROPERTIES>

```

### CCB\_SP\_TO\_MDM\_RDL - Rule Description Language

The CCB\_SP\_TO\_MDM Rule Description Language is defined as follows:

- **RDL Name:** CCB\_SP\_TO\_MDM\_RDL
- **Rule Type:** Import

### CCB\_SP\_TO\_MDM\_RDL - RDL Version

The CCB\_SP\_TO\_MDM RDL Version is defined as follows:

- **RDL Name:** CCB\_SP\_TO\_MDM\_RDL
- **Start Time:** <Date when database scripts are run>

### CCB\_SA\_TO\_MDM\_SERVICE

When service account (account data) sync requests are sent from Oracle Utilities Customer Care and Billing, they are sent to a JMS queue monitored by the CCB\_SA\_TO\_MDM\_SERVICE service. When a record is detected in the queue, the CCB\_SA\_TO\_MDM\_SERVICE business rule updates the service account data based on the details in the sync request. The CCB\_SA\_TO\_MDM\_SERVICE uses the following Runtime Service, Business Rule, and Rule Description Language.

**CCB\_SA\_TO\_MDM\_SERVICE - Runtime Service**

The CCB\_SP\_TO\_MDM Runtime Service is defined as follows:

- **Name:** CCB\_SA\_TO\_MDM\_SERVICE
- **JMS Queues:** <QUEUE\_NAME>
- **JMS Filter:** N/A
- **JMS Parameters:** N/A
- **Origination System:** ADAPTER
- **Parameters:** N/A
- **Internal Flag:** Y
- **Output Flag:** Y
- **Runtime Service Type:** QPORTAL
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME="STORAGE_LEVEL" VALUE="2" />
  <PROPERTY NAME="RULE_NAME" VALUE="CCB_SA_TO_MDM" />
  <PROPERTY NAME="INPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.XSLConverter,DB://CCB_SA_SP_SYNC_PRE" />
  <PROPERTY NAME="MAX_ERRORS" VALUE="50" />
  <PROPERTY NAME="POLL_INTERVAL" VALUE="15" />
  <PROPERTY NAME="OUTPUT_MECHANISM" VALUE="Q" />
  <PROPERTY NAME="OUTPUT_QUEUE" VALUE="[JMS Server Name]/[JMS Module
Name]![Queue Name]" />
  <PROPERTY NAME="OUTPUT_QUEUE_MSGTYPE" VALUE="T" />
  <PROPERTY NAME="OUTPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.XSLConverter,DB://
CCB_SA_SP_SYNC_SUCCESS" />
  <PROPERTY NAME="EXCEPTION_OUTPUT_MECHANISM" VALUE="Q" />
  <PROPERTY NAME="EXCEPTION_OUTPUT_QUEUE" VALUE="[JMS Server Name]/[JMS
Module Name]![Queue Name]" />
  <PROPERTY NAME="EXCEPTION_OUTPUT_QUEUE_MSGTYPE" VALUE="T" />
  <PROPERTY NAME="EXCEPTION_OUTPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.XSLConverter,DB://
CCB_SA_SP_SYNC_ERROR" />
  <PROPERTY NAME="EXCEPTION_OUTPUT_DC_1"
VALUE="com.lodestarcorp.portal.data.XSLConverter,DB://
CCB_SA_SP_SYNC_ERROR2" />
</PROPERTIES>
```

**CCB\_SA\_TO\_MDM - Business Rule**

The CCB\_SA\_TO\_MDM Business Rule is defined as follows:

- **Rule Name:** CCB\_SA\_TO\_MDM
- **Description:** Synchronizes CC&B Service Account data into MDM data repository.
- **Rule Type:** Import
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME="xmlstream" VALUE="n" />
  <PROPERTY NAME="mode" VALUE="event" />
</PROPERTIES>
```



**CCB\_SA\_TO\_MDM\_RDL - Rule Description Language**

The CCB\_SA\_TO\_MDM Rule Description Language is defined as follows:

- **RDL Name:** CCB\_SA\_TO\_MDM\_RDL
- **Rule Type:** Import

**CCB\_SA\_TO\_MDM\_RDL - RDL Version**

The CCB\_SA\_TO\_MDM RDL Version is defined as follows:

- **RDL Name:** CCB\_SA\_TO\_MDM\_RDL
- **Start Time:** <Date when database scripts are run>

**Synchronization Payload Formats**

This section describes the payload format expected by the Adapter Runtime Services and Business Rules for data synchronization between Oracle Utilities Customer Care and Billing and Oracle Utilities Meter Data Management, including examples of service point and service account (account data) requests, and a table that outlines the mapping between the XML request elements and the tables and columns each maps to in the Oracle Utilities Data Repository.

**Service Account Sync Requests - XML Format**

Service account (account data) synchronization requests use the following XML format:

```
<?xml version="1.0" encoding="UTF-8"?>
<SyncRequest xmlns="http://xmlns.oracle.com/OUMDM/RequestSync">
  <syncRequestId>1234567890</syncRequestId>
  <initialSnapshot>
    <personInfo>
      <personId>2345678901</personId>
      <name>Smith, John</name>
      <customElements/>
    </personInfo>
    <accountInfo>
      <accountId>3456789012</accountId>
      <personId>2345678901</personId>
      <customElements/>
    </accountInfo>
    <saInfo>
      <saId>4567890123</saId>
      <accountId>3456789012</accountId>
      <cisDivision>California</cisDivision>
      <saType>E-RES</saType>
      <status>Active</status>
      <saStartDate>01-JAN-2010</saStartDate>
      <saStopDate/>
      <sic>12345</sic>
      <saRateHistory>
        <effectiveDate>01-JAN-2010</effectiveDate>
        <rateSchedule>E-RES1</rateSchedule>
        <rateCode>BDrate</rateCode>
        <rateFormCode>BDrate</rateFormCode>
        <jurisCode>LODESTAR</jurisCode>
      </saRateHistory>
      <saSp>
        <saSpId>5678901234</saSpId>
        <spId>6789012345</spId>
        <saSpStartDateTime>01-JAN-2010</saSpStartDateTime>
        <saSpStopDateTime/>
        <howToUse>+</howToUse>
        <usePercent>100</usePercent>
      </saSp>
      <saSp>
        <saSpId>7890123456</saSpId>
        <spId>8901234567</spId>
```

```
<saSpStartDateTime>01-JAN-2010</saSpStartDateTime>
<saSpStopDateTime/>
<howToUse>+</howToUse>
<usePercent>100</usePercent>
</saSp>
<customElements/>
</saInfo>
<customElements/>
</initialSnapshot>
<finalSnapshot>
  <personInfo>
    <personId>2345678901</personId>
    <name>Smith, John</name>
    <customElements/>
  </personInfo>
  <accountInfo>
    <accountId>3456789012</accountId>
    <personId>2345678901</personId>
    <customElements/>
  </accountInfo>
  <saInfo>
    <saId>4567890123</saId>
    <accountId>3456789012</accountId>
    <cisDivision>California</cisDivision>
    <saType>E-RES</saType>
    <status>Active</status>
    <saStartDate>01-JAN-2010</saStartDate>
    <saStopDate/>
    <sic>12345</sic>
    <saRateHistory>
      <effectiveDate>01-JAN-2010</effectiveDate>
      <rateSchedule>E-RES1</rateSchedule>
      <rateCode>BDRate</rateCode>
      <rateFormCode>BDRate</rateFormCode>
      <jurisCode>LODESTAR</jurisCode>
    </saRateHistory>
    <saRateHistory>
      <effectiveDate>15-JAN-2010</effectiveDate>
      <rateSchedule>E-RES2</rateSchedule>
      <rateCode>BDRate</rateCode>
      <rateFormCode>BDRate</rateFormCode>
      <jurisCode>LODESTAR</jurisCode>
    </saRateHistory>
  <saSp>
    <saSpId>5678901234</saSpId>
    <spId>6789012345</spId>
    <saSpStartDateTime>01-JAN-2010</saSpStartDateTime>
    <saSpStopDateTime/>
    <howToUse>+</howToUse>
    <usePercent>100</usePercent>
  </saSp>
  <saSp>
    <saSpId>7890123456</saSpId>
    <spId>8901234567</spId>
    <saSpStartDateTime>01-JAN-2010</saSpStartDateTime>
    <saSpStopDateTime/>
    <howToUse>+</howToUse>
    <usePercent>100</usePercent>
  </saSp>
  <customElements/>
</saInfo>
<customElements/>
</finalSnapshot>
</SyncRequest>
```

## Service Account Request Elements - Table-Column Data Mapping

The table below lists the table-column mappings for the elements in the service account request XML structure.

Synchronization Request Element	MDM Table.Column	Notes
<SyncRequest>		Root element containing sync request
<syncRequestId>		Sync request ID
<initialSnapshot>		Element containing data before changes.
<personInfo>		Element containing CC&B person data
<personId>		
<name>	CUSTOMER.NAME	
<accountInfo>		Element containing account information
<accountId>	CUSTOMER.CUSTOMERID	
<personId>		
<saInfo>		Element containing service account data. Can include multipole <saSp> elements, one for each service point.
<saId>	ACCOUNT.ACCOUNTID ACCTRATECODEHIST.ACCOUNTID LSSRVCPTACCOUNT.ACCOUNTID	
<accountId>	ACCOUNT.CUSTOMERID	
<cisDivision>	ACCOUNT.JURISCODE	
<saType>	ACCOUNT.ACCOUNTTYPECODE	
<status>	ACCOUNT.ACCOUNTSTATUSCODE	
<saStartDate>	ACCOUNT.STARTTIME	
<saStopDate/>	ACCOUNT.STOPTIME	
<sic>	ACCOUNT.SIC	
<saRateHistory>		Element containing rate history information
<effectiveDate>	ACCTRATECODEHIST.STARTTIME	
<rateSchedule>		Element containing rate form/rate code
<rateCode>	RATECODE.RATECODE	
<rateFormCode>	RATEFORM.RATEFORMCODE	
<jurisCode>	RATEFORM.JURISCODE	
<saSp>		Element containing service point data
<saSpId>		
<spId>	LSSVCPTACCOUNT.SERVICEPOINTID	

Synchronization Request Element	MDM Table.Column	Notes
<saSpStartDateTime>	LSSVCPTACCOUNT.STARTTIME	
<saSpStopDateTime/>	LSSVCPTACCOUNT.STOPTIME	
<howToUse>	LSSVCPTACCOUNT.HOWTOUSE	
<usePercent>	LSSVCPTACCOUNT.USEPERCENT	
<finalSnapshot>		Element containing updated data after changes.

## Service Point Sync Requests - XML Format

Service point synchronization requests use the following XML format:

```

<SyncRequest>
  <syncRequestId>9012345678</syncRequestId>
  <initialSnapshot>
    <spInfo>
      <spId>8901234567</spId>
      <premiseId>0123456789</premiseId>
      <installDate>01-JAN-2010</installDate>
      <abolishDate/>
      <customElements/>
    </spInfo>
    <premiseInfo>
      <premiseId>0123456789</premiseId>
      <country>USA</country>
      <addressFormat>US Format</addressFormat>
      <address1>1 Main St</address1>
      <address2/>
      <address3/>
      <address4/>
      <houseType/>
      <number1/>
      <number2/>
      <inCityLimit/>
      <city>Chicago</city>
      <geographicCode/>
      <county>Cook</county>
      <state>IL</state>
      <postal>60611</postal>
      <customElements/>
    </premiseInfo>
    <customElements/>
  </initialSnapshot>
  <finalSnapshot>
    <spInfo>
      <spId>8901234567</spId>
      <premiseId>0123456789</premiseId>
      <installDate>01-JAN-2010</installDate>
      <abolishDate/>
      <customElements/>
    </spInfo>
    <premiseInfo>
      <premiseId>0123456789</premiseId>
      <country>USA</country>
      <addressFormat>US Format</addressFormat>
      <address1>1 Main Ave</address1>
      <address2/>
      <address3/>
      <address4/>
      <houseType/>
      <number1/>
      <number2/>
      <inCityLimit/>

```

```

    <city>Chicago</city>
    <geographicCode/>
    <county>Cook</county>
    <state>IL</state>
    <postal>60611</postal>
    <customElements/>
  </premiseInfo>
</customElements/>
</finalSnapshot>
</SyncRequest>

```

## Service Point Request Elements - Table-Column Data Mapping

The table below lists the table-column mappings for the elements in the service point request XML structure.

Synchronization Request Element	MDM Table.Column	Notes
<SyncRequest>		Root element containing sync request
<syncRequestId>		Sync request ID
<initialSnapshot>		Element containing data before changes.
<spInfo>		Element containing service point data
<spId>	LSSERVICEPOINT.SERVICEPOINTID	
<premiseId>	LSSERVICEPOINT.PREMISEID	
<installDate>	LSSERVICEPOINT..STARTDATE	
<abolishDate/>	LSSERVICEPOINT.STOPDATE	
<howToUse>		
<usePercent>		
<premiseInfo>		Element containing premise data
<premiseId>	LSPREMISE.PREMISEID	
<country>		
<addressFormat>	LSADDRESS.ADDRESSFORMAT	From Address Format table
<address1>	LSPREMISE.VAL#, LSADDRESS.VAL#	
<address2/>	LSPREMISE.VAL#, LSADDRESS.VAL#	
<address3/>	LSPREMISE.VAL#, LSADDRESS.VAL#	
<address4/>	LSPREMISE.VAL#, LSADDRESS.VAL#	
<houseType/>	LSPREMISE.VAL#, LSADDRESS.VAL#	
<number1/>	LSPREMISE.VAL#, LSADDRESS.VAL#	
<number2/>	LSPREMISE.VAL#, LSADDRESS.VAL#	
<inCityLimit/>	LSPREMISE.VAL#, LSADDRESS.VAL#	
<city>Chicago</city>	LSPREMISE.VAL#, LSADDRESS.VAL#	
<geographicCode/>	LSPREMISE.VAL#, LSADDRESS.VAL#	
<county>Cook</county>	LSPREMISE.VAL#, LSADDRESS.VAL#	

Synchronization Request Element	MDM Table.Column	Notes
<state>IL</state>	LSPREMISE.VAL#, LSADDRESS.VAL#	
<postal>60611</postal>	LSPREMISE.VAL#, LSADDRESS.VAL#	
<finalSnapshot>		Element containing updated data after changes.

## Configuring Synchronization Adapter Services

The Adapter services used to update service point and account data are pre-defined in the Oracle Utilities Data Repository, but must be properly configured before data synchronization can be performed. This section outlines the configuration of each of these services. See **Chapter 11: Setting Up and Configuring the Energy Information Platform Adapter** in the *Oracle Utilities Energy Information Platform User's Guide* and **Chapter 6: Setting Up, Configuring, and Running the Energy Information Platform Adapter** in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about configuring the Adapter.

**Note:** Multiple copies of each service can be active at the same time, to spread processing across servers and services. See **Configuring Bill Determinant Publisher Adapter Services** on page B-27 for more information.

### CCB\_SP\_TO\_MDM\_SERVICE

When service point sync requests are sent from Oracle Utilities Customer Care and Billing, they are sent to a JMS queue monitored by the CCB\_SP\_TO\_MDM\_SERVICE service. When a record is detected in the queue, the CCB\_SP\_TO\_MDM\_SERVICE business rule updates the service point-related data based on the details in the sync request. Configuring the CCB\_SP\_TO\_MDM\_SERVICE service involves the following:

#### Specifying the JMS Queues

The CCB\_SP\_TO\_MDM\_SERVICE service must be configured to monitor and post to the correct JMS queues when picking up sync requests from Oracle Utilities Customer Care and Billing.

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.  
The Runtime Service screen opens.
2. Enter "CCB\_SP\_TO\_MDM\_SERVICE" in the Name field and click **Search**.  
The CCB\_SP\_TO\_MDM\_SERVICE record appears on the Runtime Service screen.
3. Edit the JMS Queues field to point to the JMS queue being used by the integration.
4. Edit the Properties to specify the correct outbound/exception JMS queues (OUTPUT\_QUEUE / EXCEPTION\_OUTPUT\_QUEUE).
5. Click **Save**.

#### Enabling the Service

The CCB\_SP\_TO\_MDM\_SERVICE service must be enabled in the Service Activation table before it can be used. Use the following procedure to enable the CCB\_SP\_TO\_MDM\_SERVICE service:

1. Select **Tools and Utilities->Adapter Components->Service Activation**.  
The Service Activation screen opens.
2. Select the CCB\_SP\_TO\_MDM\_SERVICE service in the Runtime Service field and click **Search**.  
The CCB\_SP\_TO\_MDM\_SERVICE record appears on the Service Activation screen.

3. Edit the Server field to specify the server on which you wish the service to run and click **Save**.

### Creating Multiple CCB\_SP\_TO\_MDM\_SERVICE Services

When handling large numbers of sync requests, you may wish to create multiple copies of the CCB\_SP\_TO\_MDM\_SERVICE service to spread the work load across multiple services and servers. Use the following procedure to create duplicates of the CCB\_SP\_TO\_MDM\_SERVICE service:

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.  
The Runtime Service screen opens.
2. Enter “CCB\_SP\_TO\_MDM\_SERVICE” in the Name field and click **Search**.  
The CCB\_SP\_TO\_MDM\_SERVICE record appears on the Runtime Service screen.
3. Edit the Name field (to create a unique name for the copy of the service, such as CCB\_SP\_TO\_MDM\_SERVICE\_1, 2, etc.) and click **Save As New**.

**Note:** Be sure to enable each CCB\_SP\_TO\_MDM\_SERVICE service in the Service Activation table as described above.)

### CCB\_SA\_TO\_MDM\_SERVICE

When service account (account data) sync requests are sent from Oracle Utilities Customer Care and Billing, they are sent to a JMS queue monitored by the CCB\_SA\_TO\_MDM\_SERVICE service. When a record is detected in the queue, the CCB\_SA\_TO\_MDM\_SERVICE business rule updates the service point-related data based on the details in the sync request. Configuring the CCB\_SA\_TO\_MDM\_SERVICE service involves the following:

#### Specifying the JMS Queues

The CCB\_SA\_TO\_MDM\_SERVICE service must be configured to monitor and post to the correct JMS queues when picking up sync requests from Oracle Utilities Customer Care and Billing.

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.  
The Runtime Service screen opens.
2. Enter “CCB\_SA\_TO\_MDM\_SERVICE” in the Name field and click **Search**.  
The CCB\_SA\_TO\_MDM\_SERVICE record appears on the Runtime Service screen.
3. Edit the JMS Queues field to point to the JMS queue being used by the integration.
4. Edit the Properties to specify the correct outbound/exception JMS queues (OUTPUT\_QUEUE / EXCEPTION\_OUTPUT\_QUEUE).
5. Click **Save**.

#### Enabling the Service

The CCB\_SA\_TO\_MDM\_SERVICE service must be enabled in the Service Activation table before it can be used. Use the following procedure to enable the CCB\_SA\_TO\_MDM\_SERVICE service:

1. Select **Tools and Utilities->Adapter Components->Service Activation**.  
The Service Activation screen opens.
2. Select the CCB\_SA\_TO\_MDM\_SERVICE service in the Runtime Service field and click **Search**.  
The CCB\_SA\_TO\_MDM\_SERVICE record appears on the Service Activation screen.
3. Edit the Server field to specify the server on which you wish the service to run and click **Save**.

## Creating Multiple CCB\_SA\_TO\_MDM\_SERVICE Services

When handling large numbers of sync requests, you may wish to create multiple copies of the CCB\_SA\_TO\_MDM\_SERVICE service to spread the work load across multiple services and servers. Use the following procedure to create duplicates of the CCB\_SA\_TO\_MDM\_SERVICE service:

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.

The Runtime Service screen opens.

2. Enter "CCB\_SA\_TO\_MDM\_SERVICE" in the Name field and click **Search**.
3. The CCB\_SA\_TO\_MDM\_SERVICE record appears on the Runtime Service screen.
4. Edit the Name field (to create a unique name for the copy of the service, such as CCB\_SA\_TO\_MDM\_SERVICE\_1, 2, etc.) and click **Save As New**.

**Note:** Be sure to enable each CCB\_SA\_TO\_MDM\_SERVICE service in the Service Activation table as described above.

## Synchronization Adapter RDL Schemas

This section provides the XML schemas used to create the CCB\_SA\_TO\_MDM\_RDL and CCB\_SP\_TO\_MDM\_RDL Rule Description Language configurations. These can be used to extend the XML payload formats for inclusion of custom elements and attributes.

### CCB\_SA\_TO\_MDM\_RDL - Template Source XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://xmlns.oracle.com/OUMDM/RequestSync"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://xmlns.oracle.com/OUMDM/RequestSync" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="SyncRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="syncRequestId">
        </xs:element>
        <xs:element name="initialSnapshot">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="premiseInfo">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="premiseId">
                    </xs:element>
                    <xs:element name="addressFormat">
                    </xs:element>
                    <xs:element name="address1">
                    </xs:element>
                    <xs:element name="address2">
                    </xs:element>
                    <xs:element name="address3">
                    </xs:element>
                    <xs:element name="address4">
                    </xs:element>
                    <xs:element name="houseType">
                    </xs:element>
                    <xs:element name="number1">
                    </xs:element>
                    <xs:element name="number2">
                    </xs:element>
                    <xs:element name="inCityLimit">
                    </xs:element>
                    <xs:element name="city">
                    </xs:element>
                    <xs:element name="geographicCode">

```



```

        </xs:element>
        <xs:element name="county">
        </xs:element>
        <xs:element name="state">
        </xs:element>
        <xs:element name="postal">
        </xs:element>
        <xs:element name="customElements">
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="spInfo">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="spId">
            </xs:element>
            <xs:element name="premiseId">
            </xs:element>
            <xs:element name="installDate">
            </xs:element>
            <xs:element name="abolishDate">
            </xs:element>
            <xs:element name="customElements">
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
    <xs:element name="customElements" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="finalSnapshot">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="premiseInfo">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="premiseId">
                        </xs:element>
                        <xs:element name="addressFormat">
                        </xs:element>
                        <xs:element name="address1">
                        </xs:element>
                        <xs:element name="address2">
                        </xs:element>
                        <xs:element name="address3">
                        </xs:element>
                        <xs:element name="address4">
                        </xs:element>
                        <xs:element name="houseType">
                        </xs:element>
                        <xs:element name="number1">
                        </xs:element>
                        <xs:element name="number2">
                        </xs:element>
                        <xs:element name="inCityLimit">
                        </xs:element>
                        <xs:element name="city">
                        </xs:element>
                        <xs:element name="geographicCode">
                        </xs:element>
                        <xs:element name="county">
                        </xs:element>
                        <xs:element name="state">
                        </xs:element>
                        <xs:element name="postal">
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

        <xs:element name="customElements">
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="spInfo">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="spId">
            </xs:element>
            <xs:element name="premiseId">
            </xs:element>
            <xs:element name="installDate">
            </xs:element>
            <xs:element name="abolishDate">
            </xs:element>
            <xs:element name="customElements">
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="customElement" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

## CCB\_SP\_TO\_MDM\_RDL - Template Source XML Schema

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://xmlns.oracle.com/OUMDM/RequestSync"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://xmlns.oracle.com/
OUMDM/RequestSync" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="SyncRequest">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="syncRequestId">
                </xs:element>
                <xs:element name="initialSnapshot">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="personInfo">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="personId" />
                                        <xs:element name="name" >
                                        </xs:element>
                                        <xs:element name="customElements" />
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="accountInfo">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="accountId" >
                                        </xs:element>
                                        <xs:element name="personId" />
                                        <xs:element name="customElements">
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="saInfo">
                                <xs:complexType>

```

```

<xs:sequence>
  <xs:element name="saId" >
  </xs:element>
  <xs:element name="accountId" >
  </xs:element>
  <xs:element name="cisDivision" >
  </xs:element>
  <xs:element name="saType" >
  </xs:element>
  <xs:element name="status" >
  </xs:element>
  <xs:element name="saStartDate">
  </xs:element>
  <xs:element name="saStopDate">
  </xs:element>
  <xs:element name="sic" >
  </xs:element>
  <xs:element name="saRateHistory">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="effectiveDate">
        </xs:element>
        <xs:element name="rateCode">
        </xs:element>
        <xs:element name="rateFormCode">
        </xs:element>
        <xs:element name="jurisCode">
        </xs:element>
        <xs:element name="customElements">
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="saSp">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="saSpId">
        </xs:element>
        <xs:element name="spId">
        </xs:element>
        <xs:element name="saSpStartDateTime">
        </xs:element>
        <xs:element name="saSpStopDateTime">
        </xs:element>
        <xs:element name="howToUse" >
        </xs:element>
        <xs:element name="usePercent" >
        </xs:element>
        <xs:element name="customElements">
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="customElements">
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="customElements" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="finalSnapshot">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="personInfo">
        <xs:complexType>
          <xs:sequence>

```

```

        <xs:element name="personId" />
        <xs:element name="name" >
        </xs:element>
        <xs:element name="customElements">
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="accountInfo">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="accountId" >
            </xs:element>
            <xs:element name="personId" />
            <xs:element name="customElements">
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="saInfo">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="saId" >
            </xs:element>
            <xs:element name="accountId" >
            </xs:element>
            <xs:element name="cisDivision" >
            </xs:element>
            <xs:element name="saType">
            </xs:element>
            <xs:element name="status">
            </xs:element>
            <xs:element name="saStartDate">
            </xs:element>
            <xs:element name="saStopDate">
            </xs:element>
            <xs:element name="sic" >
            </xs:element>
            <xs:element name="saRateHistory">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="effectiveDate">
                        </xs:element>
                        <xs:element name="rateCode">
                        </xs:element>
                        <xs:element name="rateFormCode">
                        </xs:element>
                        <xs:element name="jurisCode">
                        </xs:element>
                        <xs:element name="customElements">
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="saSp">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="saSpId">
            </xs:element>
            <xs:element name="spId" >
            </xs:element>
            <xs:element name="saSpStartDateTime">
            </xs:element>
            <xs:element name="saSpStopDateTime">
            </xs:element>
            <xs:element name="howToUse" >
            </xs:element>
            <xs:element name="usePercent" >

```

```
        </xs:element>
        <xs:element name="customElements">
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
    <xs:element name="customElements">
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
    <xs:element name="customElement" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

## Processing Bill Determinant Requests

Oracle Utilities Customer Care and Billing uses bill determinant data (usage totals based on meter readings) calculated and stored in Oracle Utilities Meter Data Management when calculating bills for customers. This process allows Oracle Utilities Customer Care and Billing to send requests for bill determinant calculations to Oracle Utilities Meter Data Management, which in turn performs the requests calculations, and publishes the results back to Oracle Utilities Customer Care and Billing. This section describes how processing bill determinant requests from Oracle Utilities Customer Care and Billing is performed, including:

- **Overview**
- **Bill Determinant Request Adapter Services and Business Rules**
- **Calculation Request XML Format**
- **Saving Calculation Results to the Bill View Table**
- **Bill Determinant Publisher**
- **Configuring Bill Determinant Publisher Adapter Services**

### Overview

Processing of bill determinant requests from Oracle Utilities Customer Care and Billing is performed through a combination of the Oracle Utilities Meter Data Management bill determinant calculation functionality (as described in **Chapter 6: Setting Up Oracle Utilities Meter Data Management Billing Determinant Calculations**) and some additional services and functions specifically designed for integrating Oracle Utilities Meter Data Management with other systems.

At a high-level, the calculation process for requests from Oracle Utilities Customer Care and Billing includes the following steps:

1. A bill determinant calculation request is sent by Oracle Utilities Customer Care and Billing (or some other external system).
2. An Adapter service picks up the request and triggers the **Account Selector** to create a record in the **BD Request** table for the request, and a record in the **BD Queue** table for the account.
3. An Adapter service picks up the queue record and triggers the **Calculator Engine** to execute calculations for the account. Calculation rules are defined in the Oracle Utilities Rules Language.
4. Once calculations are complete, the record in the **BD Queue** table is set to Complete.
5. An Adapter service monitors the BD Queue table for Complete records and triggers the **Bill Determinant Publisher** to publish the calculation results for the account to an outbound queue.
6. Once the calculation results have been published, the record in the **BD Queue** table is moved into the **BD Queue Archive** table and its status is updated to Published.

## Bill Determinant Request Adapter Services and Business Rules

Oracle Utilities Meter Data Management uses Adapter services and business rules to handle bill determinant calculation requests from Oracle Utilities Customer Care and Billing. The descriptions of these services and rules include default settings for Runtime Service properties and Business Rule properties. See **Defining Service Properties for Adapter Runtime Services** on page 11-28 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about runtime service properties, and **Defining Properties and Parameters for Adapter Business Rules** on page 11-107 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about business rule properties.

**Note:** Multiple copies of each service can be active at the same time, to spread processing across servers and services. See **Configuring Bill Determinant Publisher Adapter Services** on page B-27 for more information.

### CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE

When bill determinant calculations are requested from Oracle Utilities Customer Care and Billing, a record is created in a JMS queue monitored by the CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE service. When a record is detected in the queue, the CCB\_INTEGRATOR\_TRIGGER\_RULE business rule triggers the Account Selector to create a record in the BD Request table and record in the BD Queue / BD Queue Archive table for the account. The CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE uses the following Runtime Service and Business Rule.

#### CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE - Runtime Service

The CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE Runtime Service is defined as follows:

- **Name:** CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE
- **JMS Queues:** <QUEUE\_NAME>
- **JMS Filter:** N/A
- **JMS Parameters:** N/A
- **Origination System:** ADAPTER
- **Parameters:** N/A
- **Internal Flag:** Y
- **Output Flag:** N
- **Runtime Service Type:** QPORTAL
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME="RULE_NAME" VALUE="CCB_INTEGRATOR_TRIGGER_RULE" />
  <PROPERTY NAME="INPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.XSLConverter,DB://CCB_INPUT"/>
  <PROPERTY NAME="STORAGE_LEVEL" VALUE="0" />
  <PROPERTY NAME="EXCEPTION_OUTPUT_MECHANISM" VALUE="Q" />
  <PROPERTY NAME="EXCEPTION_OUTPUT_QUEUE" VALUE="[JMS Server Name]/[JMS
Module Name]![Queue Name]"/>
  <PROPERTY NAME="EXCEPTION_OUTPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.XSLConverter,DB://CCB_ERR_OUTPUT"/>
  <PROPERTY NAME="EXCEPTION_OUTPUT_QUEUE_MSGTYPE" VALUE="T" />
</PROPERTIES>
```

**Note:** This service creates requests in the BD Queue table with a default BD Queue Type of "NORMAL." This queue type is inserted into the request via the "CCB\_INPUT" XMLST map (specified in the INPUT\_DC\_0 property). See **Configuring Bill Determinant Publisher Adapter Services** on page B-27 for information about how to change the BD Queue Type for requests.

**CCB\_INTEGRATOR\_TRIGGER\_RULE - Business Rule**

The CCB\_INTEGRATOR\_TRIGGER\_RULE Business Rule is defined as follows:

- **Rule Name:** CCB\_INTEGRATOR\_TRIGGER\_RULE
- **Description:** Fires the CCB integrator with the data in the payload.
- **Rule Type:** COM
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME="method" VALUE="RunCalcUsage" />
  <PROPERTY NAME="progid" VALUE="Lodestar.AccountSelector" />
  <PROPERTY NAME="xmlstream" VALUE="y" />
</PROPERTIES>
```

**BD\_QUEUEPOLLER**

The BD\_QUEUEPOLLER service monitors the **BD Queue** table and when a record is detected, triggers the **Calculator Engine** for each record. See **BD\_QUEUEPOLLER** on page 6-14 for more information about the BD\_QUEUEPOLLER service.

**BD\_QUEUEPUBLISHER**

The BD\_QUEUEPUBLISHER service monitors the **BD Queue** table and when a record is detected, triggers the **Bill Determinant Publisher** for each record. The BD\_QUEUEPUBLISHER uses the following Runtime Service.

**BD\_QUEUEPUBLISHER - Runtime Service**

The BD\_QUEUEPUBLISHER Runtime Service is defined as follows:

- **Name:** BD\_QUEUEPUBLISHER
- **JMS Queues:** N/A
- **JMS Filter:** N/A
- **JVM Parameters:** N/A
- **Origination System:** ADAPTER
- **Parameters:** N/A
- **Internal Flag:** Y
- **Output Flag:** N
- **Runtime Service Type:** BDQUEUE
- **Properties:**

```
<PROPERTIES>
  <PROPERTY NAME="POLL_INTERVAL" VALUE="5" />
  <PROPERTY NAME="POLL_COUNT" VALUE="50" />
  <PROPERTY NAME="QUEUE_TYPE" VALUE="NORMAL" />
  <PROPERTY NAME="RATE_CFG_FILE" VALUE="C:\Lodestar\cfg\Lodestar.cfg" />
  <PROPERTY NAME="PUBLISHER" VALUE="YES" />
  <PROPERTY NAME="OUTPUT_MECHANISM" VALUE="Q" />
  <PROPERTY NAME="OUTPUT_QUEUE" VALUE="[JMS Server Name]/[JMS Module
Name]![Queue Name]" />
</PROPERTIES>
```

**Note:** These are the default properties for this service. These can be changed as needed based on your requirements. See **Configuring Bill Determinant Publisher Adapter Services** on page B-27 for more information.



## Calculation Request XML Format

This section describes the payload format expected by the Adapter Runtime Services and Business Rules for bill determinant calculation requests from Oracle Utilities Customer Care and Billing.

### Calculation Request XML Format

Bill determinant calculation requests use the following XML format:

```
<request xmlns="http://xmlns.oracle.com/OUMDM/RequestBD" xmlns:fns="http://
www.w3.org/2005/xpath-functions">
  <usageId>107629479755</usageId>
  <saId>3531365813</saId>
  <createDateTime>2010-07-22T14:22:11</createDateTime>
  <startDateTime>2010-05-01T00:00:00</startDateTime>
  <endDateTime>2010-06-02T00:00:00</endDateTime>
  <dateBreaks>
    <breakDateTime>2010-05-15T00:00:00</breakDateTime>
  </dateBreaks>
  <customElements/>
</request>
```

The following table provides descriptions of the elements in the calculation request XML structure.

XML Element	Description
<request>	Root element containing a bill determinant calculation request for an account. Should contain the following XML namespace declarations: xmlns="http://xmlns.oracle.com/OUMDM/RequestBD" xmlns:fns="http://www.w3.org/2005/xpath-functions"
<usageId>	The id of the request.
<saId>	The account ID.
<createDateTime>	The date and time the request was created. This must be in the following format: "YYYY-MM-DDThh:mm:ss."
<startDateTime>	The start time of the bill period to calculate. This must be in the following format: "YYYY-MM-DDThh:mm:ss."
<endDateTime>	The stop time of the bill period to calculate. This must be in the following format: "YYYY-MM-DDThh:mm:ss."
<dateBreaks>	Element contain zero or more rate breaks between the startDateTime and endDateTime.
<breakDateTime>	The date and time on which a rate change occurred for the account. This must be in the following format: "YYYY-MM-DDThh:mm:ss."
<externalReferenceId>	ID of the batch process that created the request.
<customElements/>	Element containing zero or more custom elements.

## Saving Calculation Results to the Bill View Table

In order to publish the results of bill determinant calculations to Oracle Utilities Customer Care and Billing, you must save the calculation results in XML format in the Bill View table. This section outlines the specific data required, the XML format used for calculation results, and how to save the results to the Bill View table.

See **Appendix B: XML Statements and Functions** in the *Oracle Utilities Rules Language Reference Guide* for more information about working with XML using Oracle Utilities Rules Language.

### Required Data

Calculation results sent to Oracle Utilities Customer Care and Billing must be calculated at the service point level, and must contain the following data:

- Start Time of the bill period (BILL\_START)
- Stop Time of the bill period (BILL\_STOP)
- List of Service Point IDs for the account
  - List of bill determinants for each service point
    - For each bill determinant (referred to as a service quantity):
    - Unit of Measure for bill determinant
    - Time of Use period for bill determinant, if applicable
    - Bill Determinant Identifier Name
    - Bill determinant quantity

### Calculation Results XML Format

Bill determinant calculation results must be saved in the following XML format:

```
<MDMResponse>
  <spUsagePeriod>
    <usageStartDateTime>2010-01-14 00:00:00</startDateTime>
    <usageEndDateTime>2010-02-01 00:00:00</endDateTime>
    <serviceQty>
      <seq>10</seq>
      <spId>10000000</spId>
      <mdm_uom>KW</uom>
      <mdm_tou></tou>
      <mdm_sqi>KW</sqi>
      <qty>49.87</qty>
    </serviceQty>
    <serviceQty>
      <seq>20</seq>
      <spId>10000000</spId>
      <mdm_uom>KWH</uom>
      <mdm_tou>ON</tou>
      <mdm_sqi>KWH_ON</sqi>
      <qty>200.00</qty>
    </serviceQty>
    <serviceQty>
      <seq>30</seq>
      <spId>10000000</spId>
      <mdm_uom>KWH</uom>
      <mdm_tou>OFF</tou>
      <mdm_sqi>KWH_OFF</sqi>
      <qty>410.00</qty>
    </serviceQty>
  ...
</spUsageperiod>
<!--Summary '
  <usagePeriods>
```

```

<usageStartDateTime>2010-01-14 00:00:00</startDateTime>
<usageEndDateTime>2010-02-01 00:00:00</endDateTime>
<usageRequestType>INTERVAL</usageRequestType>
<serviceQty>
  <seq>100</seq>
  <spId></spId>
  <mdm_uom>KWH</uom>
  <mdm_tou>ON</tou>
  <mdm_sqi>KWH_ON</sqi>
  <qty>300.00</qty>
</serviceQty>
</usagePeriods>
</MDMResponse>

```

The following table provides descriptions of the elements in the calculation results XML structure.

XML Element	Description
<MDMResponse>	Root element
<spUsagePeriod>	Element containing details for the usage period.
<usageStartDateTime>	Bill Start. Must be in the following format: YYYY-MM-DDThh:mm:ss
<usageEndDateTime>	Bill Stop plus 1 second. Must be in the following format: YYYY-MM-DDThh:mm:ss
<serviceQty>	Element containing service quantity details for each service point.
<seq>	The sequence of the service quantity for the service point. The first <seq> element should be 10, and each should increment by 10. (10, 20, 30, etc.).
<spId>	The service point ID.
<mdm_uom>	The bill determinant's unit of measure
<mdm_tou>	Time of Use period for the bill determinant, if applicable
<mdm_sqi>	Bill Determinant Identifier Name (from the Bill Determinant table)
<qty>	The quantity of the bill determinant.
<usagePeriods>	Element containing summary values for all service point
<usageStartDateTime>	Bill Start. Must be in the following format: YYYY-MM-DDThh:mm:ss
<usageEndDateTime>	Bill Stop plus 1 second. Must be in the following format: YYYY-MM-DDThh:mm:ss
<usageRequestType>	The type of usage requested. Valid values are INTERVAL or SCALAR
<serviceQty>	Element containing summary values for all service points
<seq>	The sequence of the service quantity. The first <seq> element should be 10, and each should increment by 10. (10, 20, 30, etc.).
<mdm_uom>	The bill determinant's unit of measure

---

XML Element	Description
<mdm_tou>	Time of Use period for the bill determinant, if applicable
<mdm_sqi>	Bill Determinant Identifier Name (from the Bill Determinant table)

---

### Saving Results to the Database

When saving results to the Oracle Utilities Data Repository, you save the XML containing the results to the Bill View table using the SAVE TO TABLE statement. See the **Save Statements** on page 6-3 in the *Oracle Utilities Rules Language Reference Guide* for more information about this statement. When saving to the Bill View table, you must provide values for the following columns:

- ACCCOUNTID (Account ID)
- STARTTIME (Start Time of the bill period)
- BILLTIME (Start Time of the bill period)
- BILLHISTDATA (XML document containing calculation results)

**Example:** Save calculation results for the account

```
BD_RESULTS.ACCOUNTID = ACCOUNT.ACCOUNTID;  
BD_RESULTS.STARTTIME = BILL_START;  
BD_RESULTS.BILLTIME = BV.STARTTIME;  
BD_RESULTS.BILLHISTDATA = MDMRESPONSE;  
SAVE BD_RESULTS TO TABLE "BILLVIEW";
```

## Bill Determinant Publisher

When the BD\_QUEUEPUBLISHER service detects a record in the BD Queue table, it triggers the publisher which publishes the calculation results for the account to an outbound queue. For each account, the publisher does the following:

- Finds the Bill History records for the account (based on the Read Date in the BD Queue record)
- Retrieves the Bill View records for each of the account's Bill History records.
- Creates an XML document containing a combination of the calculation results from all the Bill View records, including the following elements from each Bill View record:
  - <spUsagePeriod>
  - <usagePeriods>

This XML document includes an additional element containing the IDo of the original request record (from the BD Queue record).

- Inserts the XML document into a record in the OUTBOUND\_QUEUE specified in the BD\_QUEUEPUBLISHER service.

### Publisher XML Format

The Bill Determinant Publisher publishes calculation result in the following XML format:

```
<MDMResponse>
  <usageId>234</usageId>
  <spUsagePeriod> (from 1st Bill View record)
  ...
</spUsagePeriod>
<usagePeriods> (from 1st Bill View record)
  ...
<usagePeriods>
<spUsagePeriod> (from 2nd Bill View record)
  ...
</spUsagePeriod>
<usagePeriods> (from 2nd Bill View record)
  ...
<usagePeriods>
</MDMResponse>
```

## Configuring Bill Determinant Publisher Adapter Services

The Adapter services used to publish bill determinant calculation requests are pre-defined in the Oracle Utilities Data Repository, but must be properly configured before processing can be performed. This section outlines the configuration of each of these services. See **Chapter 11: Setting Up and Configuring the Energy Information Platform Adapter** in the *Oracle Utilities Energy Information Platform User's Guide* and **Chapter 6: Setting Up, Configuring, and Running the Energy Information Platform Adapter** in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about configuring the Adapter.

### CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE

When bill determinant calculations are requested from Oracle Utilities Customer Care and Billing, a record is created in a JMS queue monitored by the CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE service. When a record is detected, the CCB\_INTEGRATOR\_JMS\_TRIGGER\_RULE business rule triggers the Account Selector to create a record in the BD Request table and record in the BD Queue / BD Queue Archive table for the account. Configuring the CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE service involves the following:

### Specifying the JMS Queues

The CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE service must be configured to monitor and post to the correct JMS queues when picking up bill determinant calculation requests from Oracle Utilities Customer Care and Billing.

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.

The Runtime Service screen opens.

2. Enter “CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE” in the Name field and click **Search**.

The CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE record appears on the Runtime Service screen.

3. Edit the JMS Queues field to point to the JMS queue being used by the integration.
4. Edit the Properties to specify the correct exception JMS queue (EXCEPTION\_OUTPUT\_QUEUE).
5. Click **Save**.

### Specifying A BD Queue Type

The CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE service creates requests in the BD Queue table with a default BD Queue Type of “NORMAL.” This queue type is inserted into the request via the “CCB\_INPUT” XSLT map (specified in the INPUT\_DC\_0 property). To change the BD Queue Type, use the following procedure.

1. Define the new BD Queue Type in the BD Queue Type table.
2. Create a copy of the “CCB\_INPUT” XSLT map defined in the Oracle Utilities Data Repository:
  - a. Select **Tools and Utilities->Adapter Components-XSLT Map Names**.  
The XSLT Map Names screen opens.
  - b. Enter “CCB\_INPUT” in the Name field and click **Search**.
  - c. Edit the Name field (to create a unique name for the copy of the record) and click **Save As New..**
  - d. Select **Tools and Utilities->Adapter Components-XSLT Map Versions**.  
The XSLT Map Versions screen opens.
  - e. Click **Add**, select the new XSLT Map Name you created above, enter a description, version numbers, and start and end dates, and click **Save**.
  - f. Select **Tools and Utilities->Adapter Components-XSLT Maps**.  
The XSLT Maps screen opens.
  - g. Select the “CCB\_INPUT” in the XSLT Map Version field and click **Search**.  
The XSLT Maps screen opens displaying the CCB\_INPUT XSLT map.
  - h. Change the XSLT Map Version field the XSLT Map Version you created above and click **Save As New**.
9. Edit the “evaluate” parameter in your copy of the XSLT map to specify the new BD Queue Type.  

```
<xsl:param name="evaluate">NEW_BD_QUEUE_TYPE</xsl:param>
```

where NEW\_BD\_QUEUE\_TYPE is the new BD Queue Type you created in step 1.
10. Click **Save**.

11. Change INPUT\_DC\_0 in the CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE service properties to point to your new XSLT Map.

### Enabling the Service

The CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE service must be enabled in the Service Activation table before it can be used. Use the following procedure to enable the CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE service:

1. Select **Tools and Utilities->Adapter Components->Service Activation**.

The Service Activation screen opens.

2. Select the CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE service in the Runtime Service field and click **Search**.

The CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE record appears on the Service Activation screen.

3. Edit the Server field to specify the server on which you wish the service to run and click **Save**.

### Creating multiple CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE services

When receiving bill determinant requests for large numbers of accounts, you may wish to create multiple copies of the CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE service to spread the work load across multiple services and servers. Use the following procedure to create duplicates of the CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE service:

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.

The Runtime Service screen opens.

2. Enter "CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE" in the Name field and click **Search**.

The CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE record appears on the Runtime Service screen.

3. Edit the Name field (to create a unique name for the copy of the service, such as CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE\_1, 2, etc.) and click **Save As New**.

**Note:** Be sure to enable each CCB\_INTEGRATOR\_JMS\_TRIGGER\_SERVICE service in the Service Activation table as described above.)

### BD\_QUEUEPOLLER

The BD\_QUEUEPOLLER service monitors the **BD Queue** table and when a record is detected, triggers the **Calculator Engine** for each record. See **BD\_QUEUEPOLLER** on page 6-22 for more information about configuring the BD\_QUEUEPOLLER service.

### Properties

When configuring the BD\_QUEUEPOLLER service when processing bill determinant calculation requests from Oracle Utilities Customer Care and Billing, you must define the following service property:

- **DISABLE\_DELETION:** Specifies that the calculation engine should NOT move the record in the BQ Queue table to the BD Queue Archive table upon completion. The only valid value for this property is "TRUE." This property should be used **ONLY** when using the BD Publisher service as part of an integration between Oracle Utilities Meter Data Management and a customer information system such as Oracle Utilities Customer Care and Billing.

## LODESTAR.CFG Settings

When configuring this service and **Calculator Engine** to process external calculation requests, include the following parameter in the LODESTAR.CFG configuration file on all application servers that will be running the calculator engine:

```
AUTO_BILL_NO_BILL_CYCLE = 1
```

If this line is present, checking for a Billing Cycle and Billing Cycle Date when processing billing for accounts is disabled. Accounts must still have an active Account History record in place, but Billing Cycle information is not needed (and ignored if present). This option should **ONLY** be used in conjunction with on-demand bill determinant calculations initiated via integration between Oracle Utilities Meter Data Management and a customer information system such as Oracle Utilities Customer Care and Billing.

## BD\_QUEUEPUBLISHER

The BD\_QUEUEPUBLISHER service monitors the **BD Queue** table and when a record is detected, triggers the **Bill Determinant Publisher** for each record. Configuring the BD\_QUEUEPUBLISHER service involves the following:

### Enabling the Service

The BD\_QUEUEPUBLISHER service must be enabled in the Service Activation table before it can be used. Use the following procedure to enable the BD\_QUEUEPUBLISHER service:

1. Select **Tools and Utilities->Adapter Components->Service Activation**.

The Service Activation screen opens.

2. Select the BD\_QUEUEPUBLISHER service in the Runtime Service field and click **Search**.

The BD\_QUEUEPUBLISHER record appears on the Service Activation screen.

3. Edit the Server field to specify the server on which you wish the service to run and click **Save**.

### Creating multiple BD\_QUEUEPUBLISHER services

When processing and publishing bill determinant requests for large numbers of accounts, you may wish to create multiple copies of the BD\_QUEUEPUBLISHER service to spread the work load across multiple services and servers. Use the following procedure to create duplicates of the BD\_QUEUEPUBLISHER service:

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.

The Runtime Service screen opens.

2. Enter "BD\_QUEUEPUBLISHER" in the Name field and click **Search**.

The BD\_QUEUEPUBLISHER record appears on the Runtime Service screen.

3. Edit the Name field (to create a unique name for the copy of the service, such as BD\_QUEUEPUBLISHER\_1, 2, etc.) and click **Save As New**.

**Note:** Be sure to enable each BD\_QUEUEPUBLISHER service in the Service Activation table as described above.



## Adjusting the Runtime Service properties

The runtime service properties of the BD\_QUEUEPUBLISHER services can be adjusted to fine-tune the performance of each service. Use the following procedure to adjust the BD\_QUEUEPUBLISHER service's properties:

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.

The Runtime Service screen opens.

2. Enter "BD\_QUEUEPUBLISHER" in the Name field and click **Search**.

The BD\_QUEUEPUBLISHER record appears on the Runtime Service screen.

3. Edit the Properties field as needed and click **Save**.

Some of the properties used by this service include:

- **POLL\_INTERVAL**: Specifies the number of seconds between polling cycles. The default is 30 seconds. To cause one or more services to poll more or less frequently, lower or raise this number as appropriate.
- **POLL\_COUNT**: Specifies the number of records to lock during each polling cycle. The default is 50. To configure one or more services to lock more or less records, raise or lower this number as appropriate. If this property is set to 0, the service will attempt to lock all of the records available.
- **QUEUE\_TYPE**: Specifies the queue type the service will look for in the BD Queue table. There is no default value for this property. If this property is not specified, the service picks up all BD Queue records, regardless of the Queue Type. To configure one or more services to monitor a specific queue, set this property accordingly. See **BD Queue Type** on page 6-4 for more information.
- **RATE\_CFG\_FILE**: Specifies the path and file name of the LODESTAR.CFG configuration file used by the Calculation Engine. The default is "C:\Lodestar\cfg\lodestar.cfg." To designate one or more services to use an alternative configuration file, set this property to the alternative file.
- **PUBLISHER**: Indicates that the service is running in Publisher mode. Valid values are YES and NO. The default is NO. Values.
- **OUTPUT\_MECHANISM**: Optional parameter for writing the outgoing data to an external target. The value can be "F", "D", or "Q" for File, Database, or Queue, respectively. No output will be produced when this parameter is not provided. If a JMS-compliant messaging system is not available, "Q" is not allowed. This property is required if OUTPUT\_DC\_0 is specified.  
**Note:** Only "Q" is supported for the Publisher.
- **OUTPUT\_DC\_0**: Optional parameter specifying the full class name of the DataConverter to use in the data output phase for non-exceptional data. Further parameters are dependent on the type of DataConverter required and should also be listed in the value attribute delimited by commas or the character specified in the DC\_DELIMITER property. Note that parameters must be specified in the order expected by the particular DataConverter. Output will be produced only when this parameter is specified. Multiple DataConverters can be "chained" together by listing each class in a separate service property with the desired ordinal number appended to the property name.
- **OUTPUT\_QUEUE**: Used in conjunction with OUTPUT\_MECHANISM being set to "Q." Specifies the name of the queue to which to write.

## Processing Replacement Readings

In the event that replacement readings for an account are received by Oracle Utilities Meter Data Management, it may be necessary to recalculate previously calculated bill determinants for that account. This process provides a mechanism for Oracle Utilities Meter Data Management to alert Oracle Utilities Customer Care and Billing upon the receipt of replacement readings. This section describes how processing of replacement reads is performed, including:

- **Replacement Reading Adapter Services and Business Rules**
- **Configuring Replacement Reading Adapter Services**

### Replacement Reading Adapter Services and Business Rules

Oracle Utilities Meter Data Management uses Adapter services and business rules to process replacement readings. The descriptions of these services and rules include default settings for Runtime Service properties and Business Rule properties. See **Defining Service Properties for Adapter Runtime Services** on page 11-28 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about runtime service properties, and **Defining Properties and Parameters for Adapter Business Rules** on page 11-107 in the *Oracle Utilities Energy Information Platform User's Guide* for more information about business rule properties.

**Note:** Multiple copies of each service can be active at the same time, to spread processing across servers and services. See **Configuring Replacement Reading Adapter Services** on page B-34 for more information.

#### CCB\_REPL\_READ

When a replacement reading is detected for an account for which bill determinants have previously been calculated, a record is created in the Payload Extension table with a Payload Type of "BDRECALC.". The CCB\_REPL\_READ service monitors this table, and when a record is detected, the CCB\_REPL\_READ service sends a message to an outbound queue containing details of the replacement reading (Account ID, Service Point ID, Start Time, Stop Time, and Meter ID). This service is only invoked if the Bill Determinant Recalculation validation has been configured for the account's meters. See **Bill Determinant Recalculation** on page 3-24 in the *Oracle Utilities Meter Data Management User's Guide* for more information. The CCB\_REPL\_READ uses the following Runtime Service.

#### CCB\_REPL\_READ - Runtime Service

The CCB\_REPL\_READ Runtime Service is defined as follows:

- **Name:** CCB\_REPL\_READ
- **JMS Queues:** N/A
- **JMS Filter:** N/A
- **JMS Parameters:** N/A
- **Origination System:** ADAPTER
- **Parameters:** N/A
- **Internal Flag:** Y
- **Output Flag:** Y
- **Runtime Service Type:** DPORTAL
- **Properties:**

```
<PROPERTIES>
<PROPERTY NAME="STORAGE_LEVEL" VALUE="0" />
<PROPERTY NAME="PAYLOAD_TYPE" VALUE="BDRECALC" />
<PROPERTY NAME="PAYLOAD_FORMAT" VALUE="C" />
```

```

<PROPERTY NAME="OUTPUT_MECHANISM" VALUE="Q" />
<PROPERTY NAME="OUTPUT_QUEUE_MSGTYPE" VALUE="T" />
<PROPERTY NAME="OUTPUT_QUEUE" VALUE="[JMS Server Name]/[JMS Module
Name]![Queue Name]" />
<PROPERTY NAME="OUTPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.XSLConverter,DB://CCB_REPL_READ" />
</PROPERTIES>

```

## CCB\_REPL\_ERROR

If there is an error when sending a notification of a replacement reading to Oracle Utilities Customer Care and Billing, a record is created in a JMS queue. The CCB\_REPL\_ERROR service monitors this queue, and when a record is detected, the CCB\_REPL\_ERROR business rule creates a work queue item noting the failure to send the message. The CCB\_REPL\_ERROR uses the following Runtime Service and Business Rule.

### CCB\_REPL\_ERROR - Runtime Service

The CCB\_REPL\_ERROR Runtime Service is defined as follows:

- **Name:** CCB\_REPL\_ERROR
- **JMS Queues:** <QUEUE\_NAME>
- **JMS Filter:** N/A
- **JMS Parameters:** N/A
- **Origination System:** ADAPTER
- **Parameters:** N/A
- **Internal Flag:** Y
- **Output Flag:** N
- **Runtime Service Type:** QPORTAL
- **Properties:**

```

<PROPERTIES>
<PROPERTY NAME="INPUT_DC_0"
VALUE="com.lodestarcorp.portal.data.XSLConverter,DB://CCB_REPL_ERROR" />
<PROPERTY NAME="STORAGE_LEVEL" VALUE="0" />
<PROPERTY NAME="RULE_NAME" VALUE="CCB_WQ" />
</PROPERTIES>

```

### CCB\_WQ - Business Rule

The CCB\_WQ Business Rule creates a work queue item indicating that the replacement usage message was not delivered. This rule is defined as follows:

- **Rule Name:** CCB\_WQ
- **Description:** Creates a work queue item.
- **Rule Type:** COM
- **Properties:**

```

<PROPERTIES>
<PROPERTY NAME="method" VALUE="CreateWQ" />
<PROPERTY NAME="progid" VALUE="Lodestar.AccountSelector" />
<PROPERTY NAME="xmlstream" VALUE="y" />
</PROPERTIES>

```

## Configuring Replacement Reading Adapter Services

The Adapter services used to process replacement readings are pre-defined in the Oracle Utilities Data Repository, but must be properly configured before processing can be performed. This section outlines the configuration of each of these services. See **Chapter 11: Setting Up and Configuring the Energy Information Platform Adapter** in the *Oracle Utilities Energy Information Platform User's Guide* and **Chapter 6: Setting Up, Configuring, and Running the Energy Information Platform Adapter** in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about configuring the Adapter.

### CCB\_REPL\_READ

When replacement usage is detected for an account for which bill determinants have previously been calculated, a record is created in the Payload Extension table with a Payload Type of "BDRECALC.". The CCB\_REPL\_READ service monitors this table, and when a record is detected, the CCB\_REPL\_READ service sends a message to an outbound queue containing details of the replacement reading (Account ID, Service Point ID, Start Time, Stop Time, and Meter ID). Configuring the CCB\_REPL\_READ service involves the following:

#### Specifying the JMS Queues

The CCB\_REPL\_READ service must be configured to post to the correct JMS queues when sending replacement reading notifications to Oracle Utilities Customer Care and Billing.

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.  
The Runtime Service screen opens.
2. Enter "CCB\_REPL\_READ" in the Name field and click **Search**.  
The CCB\_REPL\_READ record appears on the Runtime Service screen.
3. Edit the Properties to specify the correct outbound JMS queue (OUTPUT\_QUEUE).
4. Click **Save**.

#### Enabling the Service

The CCB\_REPL\_READ service must be enabled in the Service Activation table before it can be used. Use the following procedure to enable the CCB\_REPL\_READ service:

1. Select **Tools and Utilities->Adapter Components->Service Activation**.  
The Service Activation screen opens.
2. Select the CCB\_REPL\_READ service in the Runtime Service field and click **Search**.  
The CCB\_REPL\_READ record appears on the Service Activation screen.
3. Edit the Server field to specify the server on which you wish the service to run and click **Save**.

#### Creating multiple CCB\_REPL\_READ services

When processing replacement readings for large numbers of accounts, you may wish to create multiple copies of the CCB\_REPL\_READ service to spread the work load across multiple services and servers. Use the following procedure to create duplicates of the CCB\_REPL\_READ service:

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.  
The Runtime Service screen opens.
2. Enter "CCB\_REPL\_READ" in the Name field and click **Search**.  
The CCB\_REPL\_READ record appears on the Runtime Service screen.
3. Edit the Name field (to create a unique name for the copy of the service, such as CCB\_REPL\_READ\_1, 2, etc.) and click **Save As New**.

**Note:** Be sure to enable each CCB\_REPL\_READ service in the Service Activation table as described above.)

## CCB\_REPL\_ERROR

If there is an error when sending a notification of a replacement reading to Oracle Utilities Customer Care and Billing, a record is created in a JMS queue. The CCB\_REPL\_ERROR service monitors this queue, and when a record is detected, the CCB\_REPL\_ERROR business rule creates a work queue item noting the failure to send the message. Configuring the CCB\_REPL\_ERROR service involves the following:

### Specifying the JMS Queue

The CCB\_REPL\_ERROR service must be configured to monitor the correct JMS queue to pick up error messages from Oracle Utilities Customer Care and Billing.

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.  
The Runtime Service screen opens.
2. Enter “CCB\_REPL\_ERROR” in the Name field and click **Search**.  
The CCB\_REPL\_ERROR record appears on the Runtime Service screen.
3. Edit the JMS Queues field to point to the JMS queue being used by the integration and click **Save**.

### Enabling the Service

The CCB\_REPL\_ERROR service must be enabled in the Service Activation table before it can be used. Use the following procedure to enable the CCB\_REPL\_ERROR service:

1. Select **Tools and Utilities->Adapter Components->Service Activation**.  
The Service Activation screen opens.
2. Select the CCB\_REPL\_ERROR service in the Runtime Service field and click **Search**.  
The CCB\_REPL\_ERROR record appears on the Service Activation screen.
3. Edit the Server field to specify the server on which you wish the service to run and click **Save**.

### Creating multiple CCB\_REPL\_ERROR services

When processing replacement readings for large numbers of accounts, you may wish to create multiple copies of the CCB\_REPL\_ERROR service to spread the work load across multiple services and servers. Use the following procedure to create duplicates of the CCB\_REPL\_ERROR service:

1. Select **Tools and Utilities->Adapter Components-Runtime Services**.  
The Runtime Service screen opens.
2. Enter “CCB\_REPL\_ERROR” in the Name field and click **Search**.  
The CCB\_REPL\_ERROR record appears on the Runtime Service screen.
3. Edit the Name field (to create a unique name for the copy of the service, such as CCB\_REPL\_ERROR\_1, 2, etc.) and click **Save As New**.

**Note:** Be sure to enable each CCB\_REPL\_ERROR service in the Service Activation table as described above.)

# Accessing Oracle Utilities Meter Data Management from External Applications

From time to time, Oracle Utilities Customer Care and Billing users may need to view usage and other data in Oracle Utilities Meter Data Management. Accessing Oracle Utilities Meter Data Management directly from Oracle Utilities Customer Care and Billing is supported through configuration of a dashboard zone in Oracle Utilities Customer Care and Billing. This section describes how to configure access from Oracle Utilities Customer Care and Billing to Oracle Utilities Meter Data Management, including:

- **Configuring an Oracle Utilities Customer Care and Billing Dashboard Zone**
- **Accessing Oracle Utilities Meter Data Management**

## Configuring an Oracle Utilities Customer Care and Billing Dashboard Zone

Configuring access from Oracle Utilities Customer Care and Billing to Oracle Utilities Meter Data Management involves creating dashboard zone in Oracle Utilities Customer Care and Billing that displays a link that opens a new browser window and displays the Oracle Utilities Meter Data Management screens. This dashboard zone must be able to access the pass the following information from Oracle Utilities Meter Data Management to a URL that points to the Oracle Utilities Meter Data Management web application:

- Service Point ID, or
- Account ID

Refer to the Oracle Utilities Customer Care and Billing and Oracle Utilities Application Framework documentation for information about configuring a dashboard zone.

### The Oracle Utilities Meter Data Management URL

The URL that points to the Oracle Utilities Meter Data Management web application uses a specific format, comprising several elements strung together by ampersands (&), as follows:

```
http://<SERVER>/lodestar/ccs/Default.aspx?Portal=<PORTAL>&Target=<TARGET>&Mode=<MODE>&Id=<SP_ID|ACCT_ID>
```

The elements that comprise the URL are described in the table below:

URL Element	Description	Example
<SERVER>	The server name or IP address of the Oracle Utilities Meter Data Management web server	http:// <b>10.149.182.121</b> /lodestar/ccs/Default.aspx?
<PORTAL>	The portal page to display. The only supported portal option is “MDMPORTAL”	Portal= <b>MDMPORTAL</b>
<TARGET>	The target page.	Target= <b>MainWebPart.aspx</b>
<MODE>	The mode for the target page. Used when the target page is “MainWebPart.aspx”. Available options include “ServicePoint” and “Account”	Mode= <b>ServicePoint</b>
<SP_ID>	Service point ID, used when the Mode is “ServicePoint”	Id= <b>7612848283</b>
<ACCT_ID>	Account ID, used when the Mode is “Account”	Id= <b>124567</b>

Complete URLs based on the examples in the table above would be as follows:

Service Point ID:

```
http://10.149.182.121/lodestar/ccs/Default.aspx?Portal=MDMPORTAL&
Target=MainWebPart.aspx&Mode=ServicePoint&Id=7612848283
```

Account ID:

```
http://10.149.182.121/lodestar/ccs/Default.aspx?Portal=MDMPORTAL&
Target=MainWebPart.aspx&Mode=Account&Id=1234567
```

## Accessing Oracle Utilities Meter Data Management

Once the dashboard zone has been set up and is available through the Oracle Utilities Customer Care and Billing user interface, accessing Oracle Utilities Meter Data Management involves the user simply clicking the link.

- If single sign-on is enabled for Oracle Utilities Meter Data Management, when a user clicks a link that opens this URL, they will be directed to the Service Point (or Account) dashboard in Oracle Utilities Meter Data Management. See **Single Sign-On** on page 12-29 in the *Oracle Utilities Energy Information Platform Configuration Guide* for information about configuring and enabling single sign-on.
- If single sign-on is not enabled, the user will be directed to the Energy Information Platform login screen. After entering their user ID and password, they will be directed to the Service Point (or Account) dashboard.

**Note:**When the dashboard opens, the Left Menu is not displayed by default. To expand the Left Menu, move the mouse to the left side of the screen until the mouse cursor changes to a double-sided arrow and click the left mouse button and drag to the right to expand the Left Menu.





# Appendix C

---

## Setting Up Oracle Utilities Meter Data Management Web Services

This chapter describes how to set up web services used by Oracle Utilities Meter Data Management, including:

- **Oracle Utilities Meter Data Management Web Services**

# Oracle Utilities Meter Data Management Web Services

Oracle Utilities Meter Data Management web services allow Oracle Utilities Meter Data Management processes and functions to be invoked via web services, including import/export of Oracle Utilities Meter Data Management usage data. This section outlines how you setup Oracle Utilities Meter Data Management web services, and describes several pre-defined web services used with Oracle Utilities Meter Data Management.

- **Setting Up Oracle Utilities Meter Data Management Web Services**
- **Oracle Utilities Meter Data Management Business Rules**
- **Troubleshooting**
- **Request and Response XML Schemas**

## Setting Up Oracle Utilities Meter Data Management Web Services

Setting up Oracle Utilities Meter Data Management web services involves the following steps:

- **Create Web Service Records**
- **Create Action Records**
- **Create Meter Read System Records**
- **Define Read Systems and Actions in MDM.CFG.XML file**
- **Create Read System Rule Records**

### Create Web Service Records

Web Services represent individual web services. Like Runtime Services, these are the services that run on Adapter servers which execute the business rules (and associated RDLs). Records in the Web Services table contain the following fields:

- **Web Service ID:** The ID of the web service. This is used when calling the web service.
- **Description:** A description of the web service
- **Web Service Type:** The type of web service, from the Web Service Type table
- **Runtime Service:** An optional Runtime Service (from the Runtime Service table) associated with the web service
- **Business Rule:** An optional Business Rule (from the Business Rule table) associated with the web service

### Create Action Records

Records in the Action table represent meter read system actions that can be performed by Oracle Utilities Meter Data Management web services. Records in this table include the following information:

- **Action Code:** The name of the action
- **Description:** A description of the action

### Create Meter Read System Records

Records in the Meter Read Systems table represent systems used for collection of meter reads. Records in this table include the following information:

- **UID:** A unique id for the meter read system
- **Meter Read System:** The name of the meter read system
- **Description:** A description of the meter read system

## Define Read Systems and Actions in MDM.CFG.XML file

The MDM.CFG.XML configuration file defines how actions are related to meter read systems. Each read system is defined in a READSYSTEM element, and each action associated with the read system is defined in a child ACTION element. See **MDM.CFG.XML Configuration File** on page 3-5 for more information about setting up this file.

## Create Read System Rule Records

Records in the Read System Rule table associate Actions (from the Actions table) with Meter Read Systems (from the Meter Read Systems table) and Business Rules. Records in this table include the following information:

- **Read System:** The Meter Read System (from the Meter Read Systems table) which will be associated with an Action and Business Rule
- **Action Code:** The name of the action (from the Actions table) that will be associated with the Read System and Business Rule
- **Hub Rule:** The Business Rule (from the Business Rule table) that will be associated with the Read System and Action
- **Start Time:** The start time for the relationship between Read System, Action, and Business Rule
- **Stop Time:** The start time for the relationship between Read System, Action, and Business Rule

## Oracle Utilities Meter Data Management Business Rules

This section describes the pre-defined Oracle Utilities Meter Data Management business rules that can be invoked via web services. Each description includes the following:

- **Overview:** An overview of the what the business rule does.
- **Business Rule Type:** The Business Rule Type
- **Parameters:** Parameters used by the business rule
- **Properties:** Business rule properties used by the business rule
- **Request XML:** A sample request message in XML format
- **Response XML:** A sample response message in XML format

### Meter Availability Request

The Meter Availability Request rule determines if a logical meter is active and properly configured within Oracle Utilities Meter Data Management, and identifies if a logical meter is ready to receive usage.

**Business Rule Type:** MDMAvailability

#### Parameters

The Meter Availability Request business rule uses the following parameters:

Parameter	Description
ACCOUNTID	Account ID associated to the meter
SERVICEPOINTID	Service Point ID associated to the meter
SERIALNUMBER	Serial number of the physical meter
METERID	Meter ID of the meter
CHANNELID	Channel ID of the meter
UOMCODE	UOM code of the meter

One of the following is required: Account ID, Service Point ID, Serial Number, or Meter ID. If the Channel ID of the meter is not null, the Channel ID and UOM code for the meter are also required.

#### Properties

The Meter Availability Request business rule uses the following properties

Property	Description	Required
STORED_PROCEDURE_NAME	An optional stored procedure used to lookup the logical meter	No

Example:

```
<PROPERTIES>
  <PROPERTY NAME="STORED_PROCEDURE_NAME" VALUE="sp-get-meter" />
</PROPERTIES>
```

#### Request XML

```
<MDMREQUEST>
  <METERACTIVE SERIALNUMBER="SN-12345" />
</MDMREQUEST>
```

## Response XML

```
<MDMRESPONSE>
  <METERACTIVE METERID="M-XYZ" UOMCODE="01" SERIALNUMBER="SN-1234" ACTIVE="Y" />
</MDMRESPONSE>
```

## Meter Attribute Request

The Meter Attribute Request rule provides the logical and physical meter attributes of a specified meter.

**Business Rule Type:** MDMAAttribute

## Parameters

The Meter Attribute Request business rule uses the following parameters:

Parameter	Description
ACCOUNTID	Account ID associated to the meter
SERVICEPOINTID	Service Point ID associated to the meter
SERIALNUMBER	Serial number of the physical meter
METERID	Meter ID of the meter
CHANNELID	Channel ID of the meter
UOMCODE	UOM code of the meter

One of the following is required: Account ID, Service Point ID, Serial Number, or Meter ID. If the Channel ID of the meter is not null, the Channel ID and UOM code for the meter are also required.

## Properties

The Meter Attribute Request business rule uses the following properties

Property	Description	Required
STORED_PROCEDURE_NAME	An optional stored procedure used to lookup the logical meter	No

Example:

```
<PROPERTIES>
  <PROPERTY NAME="STORED_PROCEDURE_NAME" VALUE="sp-get-meter" />
</PROPERTIES>
```

## Request XML

```
<MDMREQUEST>
  <METERATTRIBUTES SERIALNUMBER="SN-12345" />
</MDMREQUEST>
```

## Response XML

```
<MDMRESPONSE>
  <METERATTRIBUTES METERID="M-12345" UOMCODE="01" SERIALNUMBER="SN-12345"
  ACTIVE="Y" NUMOFDIALS="4" METERMULTIPLIER="2" SPI="900" DCFLOW="R" BILLED="N"
  TOTALIZE="T" MODELNUM="12345" />
</MDMRESPONSE>
```

## MDM Import Request

The MDM Import Request rule loads Oracle Utilities Meter Data Management usage.

**Business Rule Type:** MDMImport

### Parameters

The MDM Import Request business rule takes usage data in the LSM format. See **Oracle Utilities Meter Data Management Web Services** on page C-2 for more information about the LSM format.

### Properties

The MDM Import Request rule uses no specific properties.

### Request XML

```
<LODESTARUSAGE>
  <MDMUSAGE METERID="MTR-12345" CHANNELID="" UOMCODE="01">
    <METERREADS>
      <METERREAD STARTREADTIME="2006-01-01T00:00:00" STOPREADTIME="2006-01-01T23:59:59" STARTVAL="36500" STOPVAL="36600" TOTALVAL="100" ENERGY="100"/>
    </METERREADS>
  </MDMUSAGE>
</LODESTARUSAGE>
```

### Response XML

```
<MDMRESPONSE>
  <METERREADIMPORT METERID="M-12345" UOMCODE="01"/>
</MDMRESPONSE>
```

## MDM Export Request

The MDM Export Request rule exports Oracle Utilities Meter Data Management usage in XML format.

**Business Rule Type:** MDMEExport

### Parameters

The MDM Export Request business rule uses the following parameters:

Parameter	Description
ACCOUNTID	Account ID associated to the meter
SERVICEPOINTID	Service Point ID associated to the meter
SERIALNUMBER	Serial number of the physical meter
METERID	Meter ID of the meter
CHANNELID	Channel ID of the meter
UOMCODE	UOM code of the meter
STARTTIME	Start Time of the usage
STOPTIME	Stop Time of the usage
USAGECATEGORY	Category (Raw, Final, or Staging)
USEEXACTDATES	Specifies that usage should be exported for the exact dates specified in the STARTTIME and STOPTIME parameters. Valid values are "Y" and "N" (default is "N").

Parameter	Description
TOTALIZE	Indicates whether or not to totalize the usage being exported. Valid values are "Y" and "N" (default is "N").
INTERVALIZE	Indicates whether or not to intervalize consumption usage being exported. Valid values are "Y" and "N" (default is "N").

The Start Time and Stop Time parameters and one of the following are required: Account ID, Service Point ID, Serial Number, or Meter ID.

### Properties

The MDM Export Request business rule uses the following properties

Property	Description	Required
STORED_PROCEDURE_NAME	An optional stored procedure used to lookup the logical meter	No

Example:

```
<PROPERTIES>
  <PROPERTY NAME="STORED_PROCEDURE_NAME" VALUE="sp-get-meter" />
</PROPERTIES>
```

### Request XML

```
<MDMREQUEST>
  <METERREADEXPORT SERIALNUMBER="SN-12345" STARTTIME="2007-01-01T00:00:00"
  STOPTIME="2007-01-31T23:59:59" />
</MDMREQUEST>
```

### Response XML (success)

```
<MDMRESPONSE>
  <METERREADEXPORT METERID="M-12345" UOMCODE="01" SERIALNUMBER="SN-12345">
    <METERREADS>
      <METERREAD STARTREADTIME="2007-01-01T00:00:00" STOPREADTIME="2007-01-
      01T23:59:59" STARTVAL="36500" STOPVAL="36600" TOTALVAL="100" ENERGY="100" />
    </METERREADS>
  </METERREADEXPORT>
</MDMRESPONSE>
```

### Response XML (failure)

```
<MDMRESPONSE>
  <METERREADEXPORTMETERID="M-12345" UOMCODE="01" SERIALNUMBER="SN-12345"
  SUCCESS="N">
    <ERRORS>
      <ERROR CODE="FAILURE" DESC="System Failure" DATE="2007-01-01T23:59:59" />
    </ERRORS>
  </METERREADEXPORT>
</MDMRESPONSE>
```

## Troubleshooting

This section describes web service error codes, and provides information regarding the cause and solution for each error code.

Error Code	Reason	Solution
FAILURE	System failed. Unexpected exception encountered that causes the system to stop processing the request.	The description of the exception will identify the type of system failure.
NODATAFOUND	No data found. The request does not link to data in the system.	Depends on the type of request. Verify the request has all required attributes to help link to data in the system.
BADCONFIG	Bad configuration. The system is not setup properly configured.	The description of the exception will identify the type of configuration problem. For example, verify the read systems in the configuration file link to valid read systems in the database and each read systems has all actions defined.
BADREQUEST	Bad request. The request received is not valid.	Depending on the type of request, verify the request has all required attributes.
BR_NOTFOUND	Business rule not found. The system was unable to identify the business rule related to the physical meter.	Relate a business rule to the physical meter and read system.
WS_FAILURE	Web Service call failed.	The description of the exception will identify the type of web service failure.
WQ_NOTFOUND	Work Queue not found.	Verify the request has the proper reference information.

## Request and Response XML Schemas

This section provides XML schemas for the Request and Response messages used by Oracle Utilities Meter Data Management business rules.

### Request Schema

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:lss="http://www.lodestarcorp.com" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:element name="MDMREQUEST">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:choice minOccurs="0">
          <xsd:element ref="METERACTIVE" maxOccurs="unbounded"/>
          <xsd:element ref="METERATTRIBUTES" maxOccurs="unbounded"/>
          <xsd:element ref="METERREADEXPORT" maxOccurs="unbounded"/>
        </xsd:choice>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```



```
</xsd:element>
<xsd:element name="METERACTIVE">
  <xsd:complexType>
    <xsd:attribute name="METERID" type="xsd:string" use="optional"/>
    <xsd:attribute name="UOMCODE" type="xsd:string" use="optional"/>
    <xsd:attribute name="CHANNELID" type="xsd:string" use="optional"/>
    <xsd:attribute name="SERIALNUMBER" type="xsd:string" use="optional"/>
    <xsd:attribute name="ACCOUNTID" type="xsd:string" use="optional"/>
    <xsd:attribute name="SERVICEPOINTID" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="METERATTRIBUTES">
  <xsd:complexType>
    <xsd:attribute name="METERID" type="xsd:string" use="optional"/>
    <xsd:attribute name="UOMCODE" type="xsd:string" use="optional"/>
    <xsd:attribute name="CHANNELID" type="xsd:string" use="optional"/>
    <xsd:attribute name="SERIALNUMBER" type="xsd:string" use="optional"/>
    <xsd:attribute name="ACCOUNTID" type="xsd:string" use="optional"/>
    <xsd:attribute name="SERVICEPOINTID" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="METERREADEXPORT">
  <xsd:complexType>
    <xsd:attribute name="METERID" type="xsd:string" use="optional"/>
    <xsd:attribute name="UOMCODE" type="xsd:string" use="optional"/>
    <xsd:attribute name="CHANNELID" type="xsd:string" use="optional"/>
    <xsd:attribute name="STARTTIME" type="xsd:date" use="optional"/>
    <xsd:attribute name="STOPTIME" type="xsd:date" use="optional"/>
    <xsd:attribute name="USAGETYPE" type="xsd:string" use="optional"/>
    <xsd:attribute name="USAGECATEGORY" type="xsd:string" use="optional"/>
    <xsd:attribute name="TOUPERIODNAME" type="xsd:string" use="optional"/>
    <xsd:attribute name="ACCOUNTID" type="xsd:string" use="optional"/>
    <xsd:attribute name="SERVICEPOINTID" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
```

## Response Schema

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:lss="http://
www.lodestarcorp.com" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xsd:include schemaLocation="MDMUSAGE.xsd"/>
  <xsd:element name="MDMRESPONSE">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:choice minOccurs="0">
          <xsd:element ref="METERACTIVE" maxOccurs="unbounded"/>
          <xsd:element ref="METERATTRIBUTES" maxOccurs="unbounded"/>
          <xsd:element ref="METERREADEXPORT" maxOccurs="unbounded"/>
          <xsd:element ref="METERREADIMPORT" maxOccurs="unbounded"/>
        </xsd:choice>
        <xsd:element ref="ERRORS" minOccurs="0"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="METERACTIVE">
    <xsd:complexType>
      <xsd:sequence minOccurs="0">
        <xsd:element ref="ERRORS"/>
      </xsd:sequence>
      <xsd:attribute name="METERID" type="xsd:string" use="required"/>
      <xsd:attribute name="UOMCODE" type="xsd:string" use="required"/>
      <xsd:attribute name="CHANNELID" type="xsd:string" use="required"/>
      <xsd:attribute name="SERIALNUMBER" type="xsd:string" use="optional"/>
      <xsd:attribute name="ACTIVE" type="xsd:string" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="METERATTRIBUTES">
    <xsd:complexType>
      <xsd:sequence minOccurs="0">
        <xsd:element ref="ERRORS"/>
      </xsd:sequence>
      <xsd:attribute name="METERID" type="xsd:string" use="required"/>
      <xsd:attribute name="UOMCODE" type="xsd:string" use="required"/>
      <xsd:attribute name="CHANNELID" type="xsd:string" use="required"/>
      <xsd:attribute name="SERIALNUMBER" type="xsd:string" use="required"/>
      <xsd:attribute name="MODELNUM" type="xsd:string" use="required"/>
      <xsd:attribute name="ACTIVE" type="xsd:string" use="required"/>
      <xsd:attribute name="NUMOFDIALS" type="xsd:string" use="required"/>
      <xsd:attribute name="METERMULTIPLIER" type="xsd:string" use="required"/>
      <xsd:attribute name="METEROFFSET" type="xsd:string" use="required"/>
      <xsd:attribute name="SPI" type="xsd:string" use="required"/>
      <xsd:attribute name="DCFLOW" type="xsd:string" use="required"/>
      <xsd:attribute name="TRANSFORMERLOSSFACTOR" type="xsd:string"
use="required"/>
      <xsd:attribute name="BILLED" type="xsd:string" use="required"/>
      <xsd:attribute name="TOTALIZE" type="xsd:string" use="required"/>
      <xsd:anyAttribute namespace="##any"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="METERREADEXPORT">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="METERREADS" minOccurs="0"/>
        <xsd:element ref="ERRORS" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="METERID" type="xsd:string" use="required"/>
      <xsd:attribute name="UOMCODE" type="xsd:string" use="required"/>
      <xsd:attribute name="CHANNELID" type="xsd:string" use="optional"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="METERREADIMPORT">
    <xsd:complexType>
      <xsd:sequence minOccurs="0">
        <xsd:element ref="ERRORS"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

```

```
</xsd:sequence>
<xsd:attribute name="METERID" type="xsd:string" use="required"/>
<xsd:attribute name="UOMCODE" type="xsd:string" use="required"/>
<xsd:attribute name="CHANNELID" type="xsd:string" use="optional"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="ERRORS">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="ERROR" maxOccurs="unbounded">
<xsd:complexType>
<xsd:attribute name="CODE" type="xsd:string" use="required"/>
<xsd:attribute name="DESC" type="xsd:string" use="required"/>
<xsd:attribute name="DATE" type="xsd:date" use="required"/>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```



---

---

# Index

## A

Account History Table 6-7  
Account Table 6-6  
Average aggregate method 4-24

## B

Bill History Value Table 4-24  
BILL\_TYPE 6-24  
BILLHISTORY Column 4-24  
Billing Cycle Date records 4-25, 6-4  
Billing Cycle records 4-25, 6-4  
Billing Modes 4-21

## C

command line program  
    MDM Estimation 5-57, 5-59

## D

Database Installation 2-9  
Database schema diagram 2-12, A-1

## E

EDI format 4-21

## I

Identifier 4-24  
Installing the Meter Data Management Software 3-2

## J

Jurisdiction field 4-25

## M

Max aggregate method 4-24  
MDM Estimation  
    command line program 5-57, 5-59  
MDM Exception Category Table 4-5  
MDM Exception Type Table 4-4  
MDM Meters 4-17  
MDM Work Queues 4-31  
Meter Configurations 4-18  
Meter Data Management Database 1-2  
Meter Data Management Overview 1-1

Meter Data Management Web components 1-2  
Meter Event Types Table 4-14  
Meter Events Table 4-14  
Meter Exception Codes Table 4-11  
Meter Manufacturers Table 4-8  
Meter Read Systems Table 4-8  
Multiple Autobill instances 5-59

## O

Operating Company Code field 4-25

## P

Physical Meter Table 4-10  
Physical Meter Types Table 4-15

## R

Read 4-16  
Read Cycle Dates Table 4-16  
Read Cycle Frequencies Table 4-16  
Read Cycle Table 4-16

## S

syntax  
    MDM estimation command line program 5-57

## T

Tables  
    Account History 6-7  
Total aggregate method 4-24  
TOU Schedules 4-2

## U

Unit of Measure 4-24  
UOM 4-24  
Usage Category Table 4-5  
Usage Exception Status Table 4-6  
Usage Reading Status Table 4-6  
Usage Type Table 4-6

## V

Validation Groups 4-17

