

Oracle Utilities Meter Data Management

Configuration Guide

Release 2.1.0 Service Pack 3

E38617-04

May 2015

Copyright © 1999, 2014 Oracle and/or its affiliates. All rights reserved.

Primary Author: Lou Proseri

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 software or related documentation 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 RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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 which 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 Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

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.

Contents

What's New

New Features in the Oracle Utilities Meter Data Management Configuration Guide	1-i
New Features for Release 2.1.0.3	1-i
New Features for Release 2.1.0.2	1-ii
New Features for Release 2.1.0.1	1-iii
New Features for Release 2.1.0.0	1-iv

Chapter 1

Overview.....	1-1
What Is This Book?.....	1-2
Other Documentation.....	1-3
Architecture Overview	1-5
Oracle Utilities Application Framework Configuration Tools.....	1-6
“Lite” Business Objects	1-6
Data Areas.....	1-6
Algorithms.....	1-6
Entity Naming Conventions	1-7
Demonstration Examples	1-8
Recommendations for Creating a Production Environment	1-8
Performance Guidelines	1-9
User Interface Guidelines	1-9
Configuration Guidelines.....	1-9
Development Guidelines	1-11
Configuration Process Overview	1-12
Basic Configuration Steps - Design Your Business Objects	1-12
Basic Configuration Steps - Create Your Business Objects	1-12
Basic Configuration Steps - Create Portals and Zones	1-13
Basic Configuration Steps - Create Master Data.....	1-13

Chapter 2

General Configuration	2-1
Installation Options	2-2
Base Time Zone	2-2
Installation Algorithms.....	2-2
Feature Configurations	2-3
Measurement Data Options	2-3
Business Intelligence Configuration.....	2-3
Master Configurations	2-4
Service and Measurement Data Foundation Master Configurations	2-4
Meter Data Management Master Configurations	2-6

Chapter 3

Setting Up Admin Data	3-1
Understanding Admin Data.....	3-2
General Admin Data	3-2

	Device Management Admin Data	3-3
	Device Installation Admin Data	3-5
	Device Communication Admin Data	3-5
	VEE Rule Admin Data	3-6
	Usage Management Admin Data	3-8
	Admin Setup Reference Tables	3-10
	Setup Sequence	3-10
	Administration Setup and Maintenance	3-10
Chapter 4		
	Devices, Measuring Components, and, Device Configurations	4-1
	Understanding Devices, Measuring Components, and Device Configurations	4-2
	Devices	4-2
	Measuring Components	4-3
	Device Configurations	4-7
	Devices In Detail	4-8
	Maintenance Object - D1-DEVICE	4-8
	Service and Measurement Data Foundation Base Package Device Business Objects	4-8
	Configuration Options	4-9
	Example Device - D1-SmartMeter	4-10
	Device Configurations In Detail	4-11
	Maintenance Object - D1-DVCCONFIG	4-11
	Service and Measurement Data Foundation Base Package Device Configuration Business Objects	4-11
11	Configuration Options	4-12
	Example Device Configuration - D1-DeviceConfiguration	4-12
	Measuring Component Types In Detail	4-14
	Maintenance Object - D1-MCTYPE	4-14
	Service and Measurement Data Foundation Base Package Measuring Component Type Business Objects	4-15
	Configuration Options	4-16
	Example Measuring Component Type - D1-IntervalChannelTypePhysical	4-17
	Meter Data Management Base Package Measuring Component Type Business Objects	4-19
	Measuring Components In Detail	4-21
	Maintenance Object - D1-MEASRCOMP	4-21
	Service and Measurement Data Foundation Base Package Measuring Component Business Objects	4-22
	Configuration Options	4-23
	Example Measuring Component - D1-IntervalChannel	4-24
	Meter Data Management Base Package Measuring Component Business Objects	4-25
	Base Package Interval Initial Measurement Functions	4-27
	Base Package Measuring Component Consumption Functions	4-28
	Configuring Devices, Measuring Components, and Device Configurations	4-29
	Configuring Custom Devices	4-29
	Configuring Custom Device Configurations	4-29
	Configuring Custom Measuring Component Types	4-29
	Configuring Custom Measuring Components	4-30
Chapter 5		
	Service Points and Device Installation	5-1
	Understanding Service Points and Device Installation	5-2
	Service Points	5-2
	Facilities and Network Locations	5-2
	Contacts	5-3
	Installation Events	5-3
	Service Providers	5-3

Measurement Cycles	5-5
Service Points In Detail	5-7
Maintenance Object - D1-SP	5-7
Service and Measurement Data Foundation Base Package Service Point Business Objects	5-7
Configuration Options	5-8
Example Service Point - D1-ServicePoint	5-9
Meter Data Management Based Package Service Point Business Objects	5-10
Facilities In Detail.....	5-11
Maintenance Object - D1-FACILITY.....	5-11
Service and Measurement Data Foundation Base Package Facility Business Objects.....	5-11
Facility-Related Extendable Lookups	5-12
Example Facility- D1-Transformer.....	5-12
Network Locations In Detail.....	5-13
Maintenance Object - D1-NWLOC	5-13
Service and Measurement Data Foundation Base Package Network Location Business Objects..	5-13
Network Location-Related Extendable Lookups	5-14
Example Network Location - D1-ElectricityNetworkLocation.....	5-14
Contacts in Detail.....	5-15
Maintenance Object - D1-CONTACT	5-15
Service and Measurement Data Foundation Base Package Contact Business Objects	5-15
Example Contact - D1-Business	5-16
Install Events in Detail	5-17
Maintenance Object - D1-INSTLEVT	5-17
Service and Measurement Data Foundation Base Package Install Event Business Objects.....	5-17
Example Install Event - D1-SmartMeterInstallEvent.....	5-18
Service Providers in Detail.....	5-20
Maintenance Object - D1-SVCPROVDR	5-20
Service and Measurement Data Foundation Base Package Service Provider Business Objects	5-20
Configuration Options	5-21
Example Service Provider - D1-HeadEndSystem	5-21
Processing Methods in Detail.....	5-23
Maintenance Object - D1-PROCMETHOD	5-23
Service and Measurement Data Foundation Base Package Business Objects.....	5-23
Configuration Options	5-24
Example Processing Method - D1-HowToCreateMCInformation.....	5-27
Meter Data Management Base Package Processing Method Business Objects and Configuration	
Options.....	5-28
Configuring Service Point and Device Installation Objects	5-30
Configuring Custom Service Points	5-30
Configuring Custom Contacts	5-30
Configuring Custom Install Events.....	5-30
Configuring Custom Service Providers	5-31
Configuring Custom Processing Methods	5-31

Chapter 6

Measurement Data.....	6-1
Understanding Initial Measurement Data and Final Measurements	6-2
Initial Measurement Data	6-2
Final Measurements	6-10
Daylight Saving Time Support.....	6-13
Understanding Estimated Initial Measurements.....	6-16
Estimated Initial Measurements	6-16
Related Measuring Component Consumption Synchronization	6-21
Overview	6-21
Understanding Consumption Synchronization Processing.....	6-23

Consumption Synchronization Configuration Settings	6-25
Configuration Example.....	6-37
Consumption Synchronization Examples.....	6-44
Initial Measurement Data In Detail.....	6-51
Maintenance Object - D1-IMD	6-51
Base Package Business Objects	6-51
Configuration Options	6-53
Example Initial Measurement - D1-InitialLoadIMDInterval	6-54
Meter Data Management Base Package Initial Measurement Data Business Objects.....	6-55
Measurements In Detail.....	6-56
Maintenance Object - D1-MSRMT	6-56
Base Package Device Business Objects	6-56
Configuration Options	6-57
Example Device - D1-Measurement	6-57
Meter Data Management Base Package Measurement Business Objects	6-57
Device Type Service Quantities in Detail.....	6-59
Maintenance Object - D1-DVTPSQ.....	6-59
Service and Measurement Data Foundation Base Package Device Type Service Quantity Business Objects.....	6-59
Configuration Options	6-60
Example Device Type Service Quantity - D1-AvgDailyEstItemConsumption	6-60
Configuring Custom Measurement Data.....	6-61
Configuring Custom Initial Measurements	6-61
Configuring Reader Remark Types	6-61
Configuring Custom Measurements	6-61
Configuring Custom Device Type Service Quantities	6-62
Chapter 7	
Validation, Editing, and Estimation	7-1
Understanding Validation, Editing, and Estimation	7-2
Overview of the Validation, Editing, and Estimation Process	7-2
VEE Rules and VEE Groups.....	7-2
VEE Roles	7-5
VEE Exceptions	7-5
VEE Groups In Detail	7-8
Maintenance Object - D1-VEEGROUP	7-8
Service and Measurement Data Foundation Base Package VEE Group Business Objects	7-8
Example VEE Group - D1-VEEGroup.....	7-9
VEE Rules In Detail.....	7-10
Maintenance Object - D1-VEERULE.....	7-10
Service and Measurement Data Foundation Base Package VEE Rule Business Objects	7-10
Configuration Options	7-11
Meter Data Management Base Package VEE Rule Business Objects.....	7-14
Base Package VEE Rules Summary	7-14
VEE Eligibility Criteria In Detail.....	7-17
Maintenance Object - D1-VEEELIGCR	7-17
Service and Measurement Data Foundation Base Package VEE Eligibility Criteria Business Objects ..	
7-17	
Configuration Options	7-18
Example VEE Eligibility Criteria - D1-VEEEligibilityCriteria	7-18
VEE Exceptions In Detail.....	7-19
Maintenance Object - D1-VEEEXCP	7-19
Service and Measurement Data Foundation Base Package VEE Exception Business Objects	7-19
Example VEE Exception - D1-VEEException	7-19
Meter Data Management Base Package VEE Exception Business Objects.....	7-21

Configuring VEE Groups, Rules, Eligibility Criteria, and Exceptions	7-22
Configuring Custom VEE Groups	7-22
Configuring Custom VEE Rules	7-22
Configuring Custom VEE Eligibility Criteria.....	7-23
Configuring Custom VEE Exceptions.....	7-23
Creating Generic Utility VEE Rules	7-23
Chapter 8	
Device Communication and Device Events.....	8-1
Understanding Activities	8-2
About Activities.....	8-2
Understanding Device Events.....	8-4
Device Events.....	8-4
Understanding Service Tasks.....	8-9
Self-Service Meter Reads.....	8-9
Service Issue Monitors and Service Investigative Orders.....	8-9
Service Task Types.....	8-12
Activities in Detail	8-14
Maintenance Object - D1-ACTIVITY	8-14
Service and Measurement Data Foundation Base Package Activity Business Objects.....	8-14
Example Activity - D1-DeviceWithDurationActivity	8-16
Meter Data Management Base Package Activity Business Objects	8-16
A Note About License Restrictions	8-17
Device Events in Detail.....	8-18
Maintenance Object - D1-DVCEVENT	8-18
Service and Measurement Data Foundation Base Package Device Event Business Objects	8-18
Example Device Event - D1-DeviceEvent	8-19
Service Tasks in Detail.....	8-21
Maintenance Object - F1-SVCTASK	8-21
Meter Data Management Base Package Service Task Business Objects.....	8-21
Example Device Event - D1-ServiceIssueMonitor	8-22
Configuring Activities and Device Event Objects	8-23
Configuring Custom Activities	8-23
Configuring Custom Device Events	8-23
Configuring Service Tasks.....	8-24
Configuring Custom Service Tasks	8-24
Configuring Service Issue Monitors and Service Investigative Orders	8-24
Chapter 9	
Usage Subscriptions	9-1
Understanding Usage Subscriptions.....	9-2
Usage Subscriptions.....	9-2
Usage Subscriptions In Detail	9-5
Maintenance Object - D1-US	9-5
Service and Measurement Data Foundation Base Package Usage Subscription Business Objects ..	9-6
Meter Data Management Base Package Usage Subscription Business Objects.....	9-6
Example Usage Subscription - D2-UsageSubscription.....	9-7
Usage Subscription Types In Detail	9-8
Maintenance Object - D1-USTYPE	9-8
Meter Data Management Base Package Usage Subscription Type Business Objects.....	9-9
Configuration Options	9-9
Example Usage Subscription Type - D2-UsageSubscriptionType.....	9-10
Configuring Usage Subscriptions and Usage Subscription Types	9-11
Configuring Custom Usage Subscriptions	9-11
Configuring Custom Usage Subscription Types	9-11

Chapter 10

Usage Groups and Usage Rules	10-1
Understanding Usage Groups and Rules	10-2
The Usage Calculation Process	10-2
Usage Rules and Usage Groups	10-2
Usage Groups In Detail	10-5
Maintenance Object - D1-USGGRP	10-5
Meter Data Management Base Package Usage Group Business Objects	10-5
Example Usage Group - D1-UsageGroup	10-6
Usage Rules In Detail	10-7
Maintenance Object - D1-USGRULE	10-7
Service and Measurement Data Foundation Base Package Usage Rule Business Objects	10-7
Meter Data Management Base Package Usage Rule Business Objects	10-8
Configuration Options	10-8
Example Usage Rule - D2-ApplyMathInt	10-9
Base Package Usage Rules Summary	10-10
Usage Eligibility Criteria In Detail	10-12
Maintenance Object - D1-USGRLELIG	10-12
Service and Measurement Data Foundation Base Package Usage Rule Eligibility Criteria Business	
Objects	10-12
Configuration Options	10-13
Example Usage Rule Eligibility Criteria - D1-UsgruleEligibilityCriteria	10-13
Meter Data Management Base Package Usage Eligibility Criteria Business Objects	10-14
Configuring Usage Rules, Groups, and Eligibility Criteria	10-15
Configuring Custom Usage Groups	10-15
Configuring Custom Usage Rules	10-15
Configuring Custom Usage Rule Eligibility Criteria	10-15
Creating Execute Usage Group Usage Rules	10-16
Creating Number Factors	10-16

Chapter 11

TOU Maps and Dynamic Options	11-1
Understanding Time of Use, TOU Maps, and Dynamic Options	11-2
Time of Use	11-2
TOU Maps	11-3
Dynamic Options and Dynamic Option Events	11-5
TOU Maps in Detail	11-6
Maintenance Object - D1-TOUMAP	11-6
Meter Data Management Base Package TOU Map Business Objects	11-6
Configuration Options	11-7
Example TOU Map - D2-TOUMap	11-7
Dynamic Options in Detail	11-8
Maintenance Object - D1-DOP	11-8
Meter Data Management Base Package Dynamic Option Business Objects	11-8
Example Dynamic Option - D2-DynamicOption	11-9
Dynamic Option Events in Detail	11-10
Maintenance Object - D1-DOPEVT	11-10
Meter Data Management Base Package Dynamic Option Event Business Objects	11-10
Example Dynamic Option - D2-DynamicOptionEvent	11-11
Configuring TOU Maps and Dynamic Options	11-12
Configuring Custom TOU Maps	11-12
Configuring Custom Dynamic Options	11-12
Configuring Custom Dynamic Option Events	11-12

Chapter 12

Usage Transactions	12-1
Understanding Usage Transactions	12-2
Usage Transactions Overview.....	12-2
Usage Transactions - A Closer Look	12-5
Creating Usage Transactions from External Systems	12-10
Sending Usage Transactions to External Systems	12-10
Usage Transactions In Detail.....	12-13
Maintenance Object - D1-USAGETRAN.....	12-13
Meter Data Management Base Package Usage Transaction Business Objects	12-14
Example Usage Transaction - D2-UsageTransaction	12-15
Configuring Usage Transactions	12-16
Configuring Custom Usage Transactions	12-16

Chapter 13

Aggregations	13-1
Understanding Aggregations	13-2
Aggregator Measuring Components and Dimensions	13-2
Types of Aggregated Data	13-2
Aggregation Calculations	13-3
Aggregator Measuring Component Types	13-6
Creating Aggregator Measuring Components	13-7
Aggregation Algorithms.....	13-8
Aggregating Specific Measuring Components.....	13-9
Configuring Aggregations	13-10
Configuring Custom Aggregations.....	13-10
Meter Data Management Base Package Aggregations	13-11

Chapter 14

Batch Processing.....	14-1
Service and Measurement Data Foundation Batch Controls	14-2
Synchronization Request Batch Controls.....	14-5
Service and Measurement Data Foundation Batch Processing Guidelines	14-6
Meter Data Management Batch Controls.....	14-9
Meter Data Management Batch Processing Guidelines.....	14-9
Meter Data Management Business Intelligence Batch Controls	14-10

Chapter 15

Integrating Oracle Utilities Meter Data Management with Other Systems	15-1
Integrating with a Customer Information System.....	15-2
Data Synchronization	15-2
Processing Usage Transaction Requests.....	15-10
Integrating with Oracle Utilities Operational Device Management	15-14
Data Synchronization	15-14
Integrating with Oracle Utilities Analytics.....	15-17
Oracle Utilities Meter Data Management Analytics Products	15-17
Oracle Utilities Meter Data Management Analytics Components	15-17
For More Information.....	15-17
Integrating with Oracle Utilities Customer Self Service	15-18
Meter Data Management Services	15-18
Configuring Rate Compare Usage Adjustment Profiles	15-19
Oracle Utilities DataConnect.....	15-25
DataConnect Data Extracts	15-25
Consumption Extracts	15-25
Master Data Extracts	15-31
Extract Flat File Formats.....	15-33
Initial Measurement and Device Event XML Formats.....	15-37

Initial Measurement Data XML Format	15-37
Device Event XML Format	15-48
Usage and Event Import - Message Driven Bean Configuration	15-53
JMS Configuration	15-53
Message Driven Bean Configuration	15-54
Notification Queue Configuration	15-55
Chapter 16	
Configuration Examples and Sample Implementation.....	16-1
Configuration Examples.....	16-2
General Data.....	16-2
Devices	16-8
Service Points and Device Installation	16-18
Device Communication	16-23
VEE Groups and Rules	16-25
Usage Groups and Rules.....	16-43
Usage Subscriptions and Usage Calculation	16-48
Sample Implementation.....	16-57
Implementation Description and Requirements.....	16-57
Implementation Steps.....	16-59
Appendix A	
Measurement Services	A-1
Appendix B	
Glossary	B-1
Appendix C	
Configuration Migration Assistant	C-1
Migration Requests.....	C-2
Base Package Migration Requests	C-2
Migration Plans	C-4
Migration Plans for Objects with CLOB-Embedded Links	C-4
Wholesale and Piecemeal Migrations	C-5
Wholesale Migrations	C-5
Piecemeal Migrations.....	C-6
Data that Cannot be Migrated Using Configuration Migration Assistant.....	C-7
Key Type	C-7
Links to System Generated IDs.....	C-7
Appendix D	
Information Lifecycle Management	D-1
ILM Components.....	D-2
ILM-Enabled Maintenance Objects.....	D-2
Maintenance Object and Business Object Options.....	D-2
Archive Eligibility	D-4
Retention Periods.....	D-4
Eligibility Summary.....	D-5
Eligibility Algorithms.....	D-6
Appendix E	
Multiple Time Zone Support	E-1
Overview.....	E-2
Date/Time Storage and Display.....	E-2
Terms and Definitions.....	E-3
Enabling Multiple Time Zone Support.....	E-4
Master Data	E-5

Overview	E-5
Device Configurations	E-5
Service Points	E-5
Usage Subscriptions.....	E-6
Install Events	E-6
Data Synchronization.....	E-7
Initial Measurement, Device Event, Usage Transaction Import	E-8
Head-End System Service Providers - Defining the Date/Time Format.....	E-8
IMD Seeder Behavior.....	E-9
Device Event Seeder	E-10
Usage Transaction Seeder.....	E-11
VEE Rules and Processing.....	E-12
Factor Processing.....	E-12
Holidays.....	E-12
TOU Map Data Generation	E-13
Overview	E-13
Validations.....	E-13
Calculating TOU Map data.....	E-14
Scalar Periodic Estimation	E-15
Outbound Messages.....	E-16
Aggregations.....	E-17
Overview	E-17
Aggregation-Related Algorithms	E-17
Smart Grid Gateway Adapters	E-18
Landis+Gyr.....	E-18
Echelon.....	E-18
MV90 Adapter for Itron	E-18
Sensus	E-19
Online Creation of Initial Measurements	E-20
Implementing Multiple Time Zones	E-21
Define Time Zones.....	E-21
Enable Multiple Time Zone Functionality.....	E-21
Create Service Providers	E-21
Create Master Data	E-22
Create TOU Data.....	E-22
Configure Data Synchronization	E-22
Configure IMD Seeder and Device Event Seeder.....	E-22
Configure Outbound Messages for Usage Transactions and Device Events	E-22
Configure SGG Adapters	E-23

What's New

New Features in the Oracle Utilities Meter Data Management Configuration Guide

This chapter outlines new features that are documented in this guide.

New Features for Release 2.1.0.3

Feature	Description	For more information, refer to...
Direct Channel on Usage Subscription	Updated description of usage subscriptions to include reference to measuring components directly related to a specific usage subscription.	Usage Subscription Attributes on page 9-2
Aggregation of Item-Based Consumption	Added information about item-based aggregation objects.	Chapter 13: Aggregations
Information Lifecycle Management	Added an appendix describing how Information Lifecycle Management (ILM) works with Oracle Utilities Meter Data Management.	Appendix D: Information Lifecycle Management
Oracle Utilities DataConnect	Added information about configuring Oracle Utilities Meter Data Management to support extract of consumption and master data for use in external systems.	Oracle Utilities DataConnect on page 15-25
New Usage Rules	Added references to the following new usage rules: <ul style="list-style-type: none">Interval Tier CalculationCheck Existence of Installed DeviceDaily Scalar Usage Rule	Meter Data Management Base Package Usage Rule Business Objects on page 10-8
ODI-Based ETL Processing	Added information about configuration details for using Oracle Data Integrator (ODI) for the extract, transform, and load (ETL) process used when integrating with Oracle Utilities Analytics.	BI Master Configuration - ODI-Based on page 2-4 BI Master Configuration - OWB-Based on page 2-4

Feature	Description	For more information, refer to...
Multiple Time Zone Support	Added an appendix describing how multiple time zones are supported in Oracle Utilities Meter Data Management.	Appendix E: Multiple Time Zone Support

New Features for Release 2.1.0.2

Feature	Description	For more information, refer to...
Configuration Examples	Added configuration examples to the Sample Implementation chapter.	Chapter 16: Configuration Examples and Sample Implementation
Support for Rate Comparisons via Oracle Utilities Customer Self-Service	Rate comparison calculations can be initiated by users of Oracle Utilities Customer Self-Service.	Self Service Master Configuration on page 2-6 Hypothetical Calculations - Rate Comparisons on page 12-8 Integrating with Oracle Utilities Customer Self Service on page 15-18
Network Location of Service Points	Facility and Network Location are new objects added to provide a means to define network locations for service points based on a related transformer (facilities) and substation/feeder (network location).	Facilities and Network Locations on page 5-2 Facilities In Detail on page 5-11 Network Locations In Detail on page 5-13
Support for Items	Items are a new type of device used to represent non-meter devices that can be involved with a customer's service. These other devices are used for many different devices including lamps, poles, current transformers, backflow devices, pulse initiators, etc.	Device Categories on page 4-2 Items on page 4-2 Items and Device Configurations on page 4-7 Service and Measurement Data Foundation Base Package Device Business Objects on page 4-8 Service Point Categories on page 5-2 Installation Events on page 5-3 Calculating Usage for Items on page 12-8
Related Measuring Component Consumption Synchronization	New consumption synchronization process allows measurement data for related measuring components to be synchronized by re-estimating existing measurements when measurement data for a related measuring component is received.	Related Measuring Component Consumption Synchronization on page 6-21
Time Zone Conversion for Incoming Initial Measurements and Device Events	Time zone translation and conversion for initial measurements and device events has been enhanced to convert the source date time information from the head-end system time zone into the target time zone	Time Zone Translation and Conversion on page 6-3 (initial measurements) Time Zone Translation and Conversion on page 8-8 (device events)

New Features for Release 2.1.0.1

Feature	Description	For more information, refer to...
Support for Scalar Periodic Estimation	Periodic estimation can be performed for scalar meters to create “estimated” initial measurements in situations when actual measurements are missing.	Understanding Estimated Initial Measurements on page 6-16
Coincident Peak Calculation via Vector and Service Quantity Math Usage Rule	The Vector & Service Quantity Math usage rule has been enhanced to provide new functions to determine the coincident maximum or minimum value of one interval vector relative to a second reference vector.	
Usage Rules Enhancements to Store Service Quantity Highlights	Several usage rules have been enhanced to store service quantity highlight date/times (such as maximum quantity date/times) and source measurement conditions in a new list introduced to the Usage Transaction.	Quality Assessment and Bill Print Options on page 12-6
Usage Rules Enhancements to Perform Measurement Quality Assessment	Several usage rules have been enhanced to examine the quality of the source measurement data and store this information in a new list added to the Usage Transaction, and to populate a new flag within the Usage Transaction to indicate whether the quantity is based on data that is not the highest quality.	Quality Assessment and Bill Print Options on page 12-6
Usage Transaction Export Enhancements for Oracle Utilities Customer Care and Billing	<p>Several enhancements have been made to support bill print requirements via outbound messages used to send usage transaction service quantity information to Oracle Utilities Customer Care and Billing. These include</p> <ul style="list-style-type: none"> • Device Identifier and Service Point Information • Highlight Date/Times (maximum/minimum quantity date/times) associated with usage transaction service quantities • Estimation indicators 	Quality Assessment and Bill Print Options on page 12-6
Configuration Migration Assistant: Sample Migration Plans	The Oracle Utilities Application Framework has introduced functionality to permit export and import of administrative data between environments.	<p>Appendix C: Configuration Migration Assistant</p> <p>Refer to the Oracle Utilities Application Framework documentation for more information.</p>

New Features for Release 2.1.0.0

Feature	Description	For more information, refer to...
Support for Reader Remarks	Reader remarks capture and/or record specific events or circumstances encountered when a meter reader is manually reading scalar meters.	Reader Remarks and Scalar Measurements on page 6-8 Configuring Reader Remark Types on page 6-61 Initial Measurement Data XML Format on page 15-37 Example - Scalar Initial Measurement Data on page 15-38
Support for Service Investigative Orders	Service investigative orders are activities created when a specified set of events have occurred at a service point.	Service Issue Monitors and Service Investigative Orders on page 8-9 About Service Issue Monitor Types on page 8-12 Configuring Service Issue Monitors and Service Investigative Orders on page 8-24
Support for “master” and “sub” usage subscriptions.	"Master" and "sub" usage subscriptions allow for hierarchical relationships between usage subscriptions.	Usage Subscription Attributes on page 9-2
New Override Start and End Date/Time fields for usage subscription service points.	Override dates and times are used to specify the start/end date and time when identifying measurements used to calculate usage for the usage subscription.	Usage Subscription Attributes on page 9-2
Support for sending an interval data snapshot along with usage transaction information.	Usage transactions can be configured to include summary service quantities and an interval data “snapshot” along with service quantities and other information sent to external systems.	Including Summary Service Quantities and Interval Data Snapshots with Usage Transactions on page 12-11

Chapter 1

Overview

This chapter provides an overview of this configuration guide and an introduction to the Oracle Utilities Meter Data Management application. This includes:

- **What Is This Book?**
- **Architecture Overview**
- **Oracle Utilities Application Framework Configuration Tools**
- **Demonstration Examples**
- **Recommendations for Creating a Production Environment**
- **Performance Guidelines**
- **Configuration Process Overview**

What Is This Book?

This guide describes how to configure Oracle Utilities Meter Data Management. It is intended for implementers and system administrators responsible for configuration and initial setup of the application.

Oracle Utilities Meter Data Management is based on the Oracle Utilities Application Framework (OUAF). For information about using and configuring basic Framework functions, see the Oracle Utilities Application Framework documentation. This guide only covers configuration of functions specific to Oracle Utilities Meter Data Management.

The body of this guide presents conceptual information to help you understand how the system works as well as how the various configuration options affect system functionality. Once you have an understanding of the system's capabilities, you can plan your data setup and design any customizations you want to implement.

When you are ready to implement your design, use the Admin Setup Reference Tables in Chapter 3 to guide you through the setup process of admin data. This section lists each object that can be configured, defines any prerequisites for configuration.

Note: The sequence in which you configure system objects is very important. Admin Setup Reference Tables describes admin data dependencies and defines the order in which admin objects should be configured. By following this sequence carefully, you can streamline the configuration process and reduce the amount of time required for setup.

This guide includes the following chapters:

- **Chapter 1: Overview** (this chapter) provides an overview of the Oracle Utilities Meter Data Management architecture and of the configuration tools and process used in implementing the product.
- **Chapter 2: General Configuration** provides an overview of some general configuration options used by the system.
- **Chapter 3: Setting Up Admin Data** describes the different types of admin data that must be set up and defined as part of implementing and configuring Oracle Utilities Meter Data Management.
- **Chapter 4: Devices, Measuring Components, and, Device Configurations** provides an overview of devices and measuring components and how they are used in the system, along with technical details concerning device-related maintenance and business objects.
- **Chapter 5: Service Points and Device Installation** provides an overview of service points and device installation-related objects and how they are used in the system, along with technical details concerning related maintenance and business objects.
- **Chapter 6: Measurement Data** provides an overview of initial and final measurement data and how it is used in the system, along with technical details concerning related maintenance and business objects.
- **Chapter 7: Validation, Editing, and Estimation** provides an overview of the validation, editing, and estimation process, along with technical details concerning related maintenance and business objects.
- **Chapter 8: Device Communication and Device Events** provides an overview of the device communication-related objects and how they are used in the system, along with technical details concerning related maintenance and business objects.
- **Chapter 9: Usage Subscriptions** provides an overview of usage subscriptions and how they are used in the system, along with technical details concerning related maintenance and business objects.

- **Chapter 10: Usage Groups and Usage Rules** provides an overview of the usage calculation process, along with technical details concerning related maintenance and business objects.
- **Chapter 11: TOU Maps and Dynamic Options** provides an overview of TOU maps and dynamic options and how they are used in the system, along with technical details concerning related maintenance and business objects.
- **Chapter 12: Usage Transactions** provides additional information about the usage calculation process, along with technical details concerning usage transaction-related maintenance and business objects.
- **Chapter 13: Aggregations** provides information about how aggregation is performed by the application.
- **Chapter 14: Batch Processing** provides a list of the base package batch controls provided with the system and guidelines for batch processing.
- **Chapter 15: Integrating Oracle Utilities Meter Data Management with Other Systems** provides information about integrations between Oracle Utilities Meter Data Management and external systems, including Oracle Utilities Customer Care and Billing, Oracle Utilities Operational Device Management, Oracle Utilities Business Intelligence, Oracle Utilities DataConnect, and Oracle Utilities Customer Self-Service.
- **Chapter 16: Configuration Examples and Sample Implementation** provides a high-level description of the steps involved in configuring Oracle Utilities Meter Data Management in a simple example implementation.
- **Appendix A: Measurement Services** provides a list of base package measurement services use by VEE rules and functions.
- **Appendix B: Glossary** is a list of commonly used terms.
- **Appendix C: Configuration Migration Assistant** provides information about how to use the Configuration Migration Assistant feature of the Oracle Utilities Application Framework with Oracle Utilities Meter Data Management.
- **Appendix D: Information Lifecycle Management** provides information about how to use the Information Lifecycle Management feature of Oracle Database with Oracle Utilities Meter Data Management.
- **Appendix E: Multiple Time Zone Support** provides information about support for multiple time zones in Oracle Utilities Meter Data Management.

Other Documentation

This section describes other documentation provided with Oracle Utilities Meter Data Management.

Installation Documentation

Installation documentation describes the steps involved in the installation and initial set up of the system, and includes the following documents:

- Oracle Utilities Meter Data Management Quick Install Guide
- Oracle Utilities Meter Data Management DBA Guide
- Oracle Utilities Meter Data Management Installation Guide

User Documentation

User documentation provides conceptual information and procedures related to working with the various objects used in the system, and includes the following documents:

- Oracle Utilities Application Framework Business Process Guide
- Oracle Utilities Application Framework Administration Guide
- Oracle Utilities Service and Measurement Data Foundation User's Guide
- Oracle Utilities Meter Data Management User's Guide

Supplemental Documentation

Supplemental documentation provides technical information related to system administration tasks and include the following documents:

- Oracle Utilities Meter Data Management Server Administration Guide
- Oracle Utilities Meter Data Management Batch Server Administration Guide

Embedded Help

Oracle Utilities Meter Data Management, like all Oracle Utilities Application Framework applications, provides extensive internal documentation. For example, detailed descriptions of system objects are included in the objects' maintenance portals. The lifecycle of each business object is described on the Lifecycle tab and depicted in flow diagrams on the Summary tab. This information is extremely useful for implementers and system administrators.

Embedded help is provided for all non-obvious fields in most portals and zones. If a field has associated help text, a ? icon appears next to the field when the zone is displayed.

Online Help

Oracle Utilities Meter Data Management also includes context-sensitive help for all the user interface screens users will typically work with as they use the system. Online help contains conceptual information and procedures related to working with the various objects used in the system.

The online help is divided into the following three sections:

- Oracle Utilities Application Framework: Describes the features and functions of the application framework (F1)
- Oracle Utilities Service and Measurement Data Foundation: Describes the features and functions provided in the Service and Measurement Data Foundation (D1)
- Oracle Utilities Meter Data Management: Describes the features and functions provided in the meter data management application (D2)

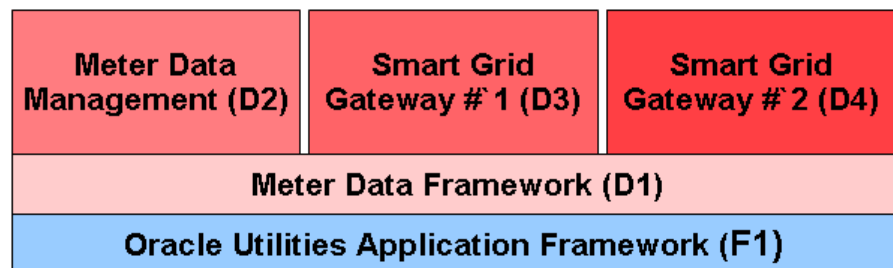
Architecture Overview

Oracle Utilities Meter Data Management is used to maintain information about meters and the service points at which they are installed. The application provides means of recording measurements and events associated with meters in the field as well as the ability to compute usage for the recorded measurements.

Oracle Utilities Meter Data Management comprises the following functional areas:

- **Device Management:** the maintenance of physical meters in the field
- **Device Installation:** the maintenance of service points and the installation of meters in the field. This includes the means of registering outside systems to Oracle Utilities Meter Data Management for provider/consumer-specific processing of meter events and activities
- **Device Communication:** the maintenance of communications between Oracle Utilities Meter Data Management and head-end systems, including import of usage and events.
- **Validation, Editing, and Estimation:** the maintenance of measurement data and the engine used to validate and modify that data as it comes in
- **Usage Management:** the engine that calculates billable usage recorded on devices, applying factors and dividing the usage into configurable time of use periods
- **Aggregation:** summarization of measurement data for reporting and analysis purposes
- **Data Synchronization:** synchronizing data between Oracle Utilities Meter Data Management and other Oracle Utilities applications (including Oracle Utilities Operational Device Management and Oracle Utilities Customer Care and Billing).
- **Oracle Utilities DataConnect:** extraction of consumption and master data for use in external systems.

Oracle Utilities Meter Data Management is built upon the Oracle Utilities Service and Measurement Data Foundation, a framework that provides shared functionality used by Oracle Utilities Meter Data Management, Oracle Utilities Smart Grid Gateway, and other Oracle Utilities products. Oracle Utilities Meter Data Management and the Oracle Utilities Service and Measurement Data Foundation are built atop the Oracle Utilities Application Framework.



Oracle Utilities Application Framework Configuration Tools

The Oracle Utilities Application Framework (OUAF) configuration tools can be used to create and customize system entities, such as business objects, portals, zones, and UI maps. Refer to the Oracle Utilities Application Framework configuration tools documentation for instructions on using these tools.

This configuration guide does not duplicate the concepts and procedures presented in the Oracle Utilities Application Framework configuration tools documentation; rather, it will identify the specific objects used by Oracle Utilities Meter Data Management that can be configured and customized using the configuration tools, as well as application parameters and objects that can be managed within the application components themselves.

This guide assumes that all individuals responsible for system configuration and implementation will be familiar with the Oracle Utilities Application Framework and will have completed training on the Oracle Utilities Application Framework Configuration Tools.

The following sections discuss some specific topics related to the configuration tools.

“Lite” Business Objects

When a business object is read, the Oracle Utilities Application Framework dynamically constructs a SQL statement to retrieve the rows and columns associated with the business object's schema. If a process only needs only a small subset of a business object's elements, a “lite” business object that only references these elements can be used.

These “lite” business objects are used by the business processes (typically the construction of info strings) that only need a small subset of elements. Lite business objects are configured to never allow instances. In other words, they are only used to read existing instances of other business objects.

Later chapters in this book list the various “lite” business objects provided with the base package for each of the types of business objects described (devices, measuring components, service points, etc.)

Data Areas

As described in the Oracle Utilities Application Framework documentation, data areas provide a common schema location for re-used schema structures. Data areas exists solely to help eliminate redundant element declaration. For example, if you have multiple schemas that share a common structure, you can set up a stand-alone data area schema for the common elements and then include it in each of the other schemas.

Many of the base package schemas make use of use data areas, and Oracle recommends that you take advantage of data areas where possible to avoid redundant data definition.

Algorithms

Many functions in the system are performed using user-defined algorithms (also referred to as plug-ins). For example, user-defined algorithms can be used to perform custom validation, editing, and estimation logic, or retrieve characteristic values for factors.

Custom algorithms allow implementers to modify how the system responds to certain system events. The system provides system where the custom algorithms can be invoked instead of the base package algorithms provided with the system. For instructions on creating custom algorithms and algorithm types, see the Oracle Utilities Application Framework documentation. To view information about specific algorithms provided with the base system, use the Application Viewer (also described in the Oracle Utilities Application Framework documentation). The Application Viewer provides information about the base logic, inputs, and outputs of each algorithm entity or plug-in spot.

Entity Naming Conventions

Oracle Utilities Meter Data Management system uses naming conventions to identify and distinguish entities that belong to different Oracle applications. These conventions can help you locate entities and understand their context.

Each base product uses a 2-character owner code as a prefix for all its entities. For Oracle Utilities Meter Data Management, these prefixes are as follows:

- All Oracle Utilities Application Framework entities start with "F1"
- All Oracle Utilities Service and Measurement Data Foundation entities start with "D1"
- All Oracle Utilities Meter Data Management entities start with "D2"

Oracle recommends that you follow these naming conventions and develop your own set of conventions for the entities you create. If you create new entities, DO NOT use these prefixes; use the prefix "CM" (or some other unique prefix) to identify entities that have been customized.

Demonstration Examples

The demonstration environment shipped with the base product provides setup examples that may be helpful as you implement your system.

Recommendations for Creating a Production Environment

Oracle recommends that you do not clone the demonstration environment as a basis for a new production environment. The demonstration environment typically includes transactional data that will be irrelevant to your production environment and can cause unexpected issues if it is not purged correctly. The recommended process is to start a new production environment from a new installation and migrate “clean” system data (such as business objects and algorithms) and administrative data (such as sample activity types or other administrative entities) from the demonstration and/or test or development environments as applicable.

Your implementation can use bundling and/or the Configuration Migration Assistant to move system and administrative data. Instructions for using these tools are provided in detail in the chapters titled “Bundling” and “Configuration Migration Assistant” in the *Oracle Utilities Application Framework Administration Guide*. For more information about base package Oracle Utilities Meter Data Management migration requests that can be used when migrating administrative data, refer to **Appendix C: Configuration Migration Assistant**.

Please contact Oracle customer support for assistance.

Performance Guidelines

This section provides a number of performance-related guidelines and recommendations to take into account when implementing Oracle Utilities Meter Data Management. This includes guidelines related to the following:

- **User Interface Guidelines**
- **Configuration Guidelines**
- **Development Guidelines**

User Interface Guidelines

Zones

For better performance, user interface zones should be initially collapsed. The initial state of zones (collapsed or not) can be controlled via the "Portal Preferences" tab on the **User** portal. The number of records returned to the user interfaces should be limited to 25 rows.

Configuration Guidelines

This section outlines configuration guidelines and recommendations for improved performance related to the following:

- **Importing Initial Measurements and Device Events**
- **VEE Processing**
- **Usage Transaction Processing**
- **Consumption Synchronization Processing**

Importing Initial Measurements and Device Events

When importing initial measurements and device events through Oracle Service Bus, Oracle Utilities recommends using import files containing an average of 1,000 per file.

- Lower number of devices per file can require more time for processing.
- Higher numbers of devices per file can lead to high growth in collection of bad data that can result in lower throughput.
- The optimum number of transactions per file may vary by head-end system.

Initial measurement payloads should have selective criteria to identify the exact measuring component for each measurement. Where possible this should include the measuring component identifier along with device serial number. If this is not provided, the UOM/TOU/SQI configured on the measuring component type is used to retrieve the exact business object from the Service Provider. This can be problematic in cases where the same UOM/TOU/SQI values are used for multiple measuring components.

VEE Processing

High-Low Check

The High-Low Check VEE rule is an input/output (I/O) intensive operation. Lower values for the "Historic Pre-Window" parameter can help improve performance of this rule.

VEE Processing Relative Costs

Each VEE rule included in VEE processing can have an impact on overall performance. The table below summarizes the relative costs in terms of performance for several commonly used VEE rules.

Using this table when designing your VEE groups and rules can help identify potential performance challenges.

For example, using "Minimal Impact" rules will likely have little to no significant impact on performance, while using several "Moderate Impact" rules could result in performance challenges, and using the High Low Check rule is very likely to introduce noticeable impacts to VEE processing performance.

Relative Cost	VEE Rules
Minimal Impact (0% - 5%)	Interval Interpolation Interval Replacement Rule Interval Size Validation Interval Spike Check Negative Consumption Check Scalar Replacement Rule UOM Check
Moderate Impact (5% - 20%)	Final Measurement Validation Interval Adjustment From Scalar Interval Averaging Multiplier Check Scalar Calculation From Interval Scalar Proration Sum Check
Significant Impact (20%)	High/Low Check

Usage Transaction Processing

Saving derived interval data and sending the data to an external system via the "Save Interval Vector" and "Extract Interval Data" options on the "Vector and Service Quantity Math" and "Get Interval Data" Usage rules can have a noticeable impact on usage transaction processing. Usage transaction processing can be improved by up to 45% if extraction of interval data is eliminated. Use of these options should be limited to situations in which they are absolutely required.

Consumption Synchronization Processing

Consumption synchronization processing should only be run on devices where the expected difference in consumption is outside the defined tolerance as dictated by the Sum Check VEE rule. In a production environment, Oracle Utilities advises running consumption synchronization for two percent of devices or less.

Development Guidelines

The following are recommended guidelines for creating custom ("CM") entities and related algorithms.

1. Information Algorithms should be designed at the maintenance object level to avoid the need to invoke business objects.
2. Avoid making unbounded SQL statements with no boundary condition on date columns.
3. Using of Scripts for simple VEE rules is approximately 10% more expensive than a similar Java rule.
4. Reduce the number of searches on information stored in CLOB columns, and use physical columns wherever possible.
5. Implement caching to pre-fetch data instead of issuing multiple SQL statements.
6. For Outbound Messages, the schema extraction for usage transactions should be converted into a RAW string operation to allow for more efficient Pre-Processing logic.
7. If direct updates/inserts to database are sufficient and different element level Validation, Pre-Processing, and Post-Processing logic not required, use database entities for retrieving and updating records instead of creating business object instances, and avoid creating COTSInstances (which internally trigger element validations). For update/inserts, use database entities for work fields and Raw-based business objects for non-work fields.
8. UI Map schemas should be designed to display specific, required data. This will ensure the application will only retrieve the elements that are required for display.

Configuration Process Overview

This section provides a high-level overview of some of the steps involved in the configuration process when implementing Oracle Utilities Application Framework products such as Oracle Utilities Meter Data Management.

Note: The following sections are a simplification of an involved process, and are provided as guidelines only. Refer to the Oracle Utilities Application Framework documentation for detailed information about business objects and other objects referenced below.

Basic Configuration Steps - Design Your Business Objects

Much of the configuration involved in implementing Oracle Utilities Meter Data Management is centered around the creation of business objects. Nearly every object or set of data used by Oracle Utilities Meter Data Management is defined in business object, including meters and registers (devices and measuring components), service points, contacts, measurement data, validation rules, usage calculation rules, and more.

Given the prominent role that business objects play, one of the most important steps in implementing Oracle Utilities Meter Data Management is identifying the business objects you will need to create to meet the requirements of your implementation. At a high level, this includes the following steps:

1. Identify the data to be defined by each business object

This step defines the maintenance object to be used with the business object, and the data elements to include in the business object's schema. Leverage data areas where possible to minimize redundant data definition.

2. Identify the processing to be performed by each business object

This step determines the specific algorithms/algorithm types, and business services (and related scripts and service programs) that will be perform the processing required by your business objects.

3. Identify how users will access and work with each business object (if applicable)

This defines the portals, zones, navigation options, BPA scripts, etc. you will need to develop to allow users access to your business objects.

Basic Configuration Steps - Create Your Business Objects

After identifying the above information, the next step is to create the business objects used in your implementation. At a high level, this includes the following steps:

1. Create configuration objects

Before you can create your business objects, you must first create the various configuration objects used by each business object, including:

- Application Services
- UI Maps (display and maintenance, as needed)
- Navigation Options
- Service Scripts
- Algorithm Types/Algorithms
- BPA Scripts/Business Service
- Other business objects
- Etc.

Where possible, leverage base package objects instead of creating your own to minimize data redundancy.

2. Create the business object

Once the configuration objects used by the business object are in place, you can create the actual business object itself using the Business Object portal, referencing the configuration objects created in step 1 as appropriate.

Later chapters in this book provide examples of many of the base package business objects provided with the system. These are provided to illustrate how the base package objects were designed, and to serve as the basis for the business objects you create as part of your implementation.

Basic Configuration Steps - Create Portals and Zones

If the base package portals and zones are not sufficient to meet the requirements of your implementation, you may have to create your own to allow users to work with your business objects. This can include creating the following:

- Context Menus
- Menus and Menu Items
- Navigation Keys
- Navigation Options
- Portals
- Zones

Basic Configuration Steps - Create Master Data

The “master” data used by Oracle Utilities Meter Data Management includes the various entities used in your implementation, such as devices, measuring components, service points, VEE rules, etc. This data must be created in the system before you can process measurement data and create bill determinants from the data. Creating this data includes the following steps:

1. Create admin “type” data.

Many of the objects used by Oracle Utilities Meter Data Management have corresponding admin “type” objects that are used to define attributes common to instances of that type of object. For example, Device Types are used to define attributes common to devices of a specific type. One of the most important attributes defined by an admin “type” object is the business object that will be used for instances of the object of that type. For example, devices created from a Device Type that references the “D1-SmartMeter” device business object will be based on that business object.

The **Admin “Type” Objects** table below lists the core objects used by Oracle Utilities Meter Data Management and their corresponding admin “type” objects.

2. Create instances of the data.

Once the admin “type” data is in place, you can create the instances of the master data objects used in your implementation. These instances are the individual devices, measuring components, service points, VEE rules, usage subscriptions, etc. that will be used in processing measurement data, calculating usage and bill determinants, and so on.

Admin “Type” Objects.

Object	Admin “Type” Object
Activity	Activity Type
Consumption Extract Request	Request Type Consumption Extract Type
Contact	Contact Type
Device	Device Type
Device Configuration	Device Configuration Type
Device Event (Reader Remark)	Device Event Type (Reader Remark Type)
Dynamic Option	Dynamic Option Type
Measuring Component	Measuring Component Type
Service Point	Service Point Type
Service Task	Service Task Type
TOU Map	TOU Map Type
Usage Subscription	Usage Subscription Type

Chapter 2

General Configuration

This chapter describes configuration of general components, including the following:

- **Installation Options**
- **Feature Configurations**
- **Master Configurations**

Installation Options

Installation options define the individual applications installed on your system and identify algorithms used to implement core system functions. These options also define global parameters such as the administrative menu style (alphabetical or functional), the country, language, currency code, as well as the base time zone to use for this implementation.

Installation options are stored in the installation record for your system. Use the **Installation Options - Framework** portal to configure these options. This portal is part of the OUAF and is described in detail in the Framework documentation.

Base Time Zone

The date/time attributes for all time-sensitive application entities, including start and end dates, are stored in the server application time zone in standard time and displayed in that time zone's legal time, which is the standard time adjusted for any seasonal shift.

The server time zone, also referred to as the Base Time Zone, must be correctly specified on the installation options record.

Note: The installation record does not dictate the server time zone, but rather must match it.

Installation Algorithms

Installation algorithms implement global system functions and can be customized for each implementation. The base package supports the following installation options for Meter Data Management-related system events:

- **Geocoding Service:** Responsible for geocoding an address (converting an address to a geocode latitude/longitude pair).
- **Global Context:** Sets global contexts (displayed in the Global Context dashboard zone) based on the value of existing global contexts. For example, if the Service Point is specified, this algorithm sets the Device by finding the most recently installed Device on the service point. It then sets the Measuring Component by finding the most effective Device Configuration and retrieving any measuring component linked to it. It then sets the Usage Subscription by finding the most recent active usage subscription linked to the service point. The contact is set by finding the main contact for the usage subscription.

Feature Configurations

Some of the features in Oracle Utilities Application Framework based applications are configured by populating options on a "feature configuration". For example, if your implementation uses Oracle Utilities Customer Care and Billing's batch scheduler, you must populate a variety of options on the batch scheduler's feature configuration.

Refer to the *Oracle Utilities Application Framework Administration Guide* for more information about defining feature configurations.

Oracle Utilities Meter Data Management uses the following types of feature configurations:

Measurement Data Options

Measurement Data Options are used to define behavior related to periodic estimation of initial measurement data, including:

- **No of Hours in Past to Retrieve Last Usable Measurement:** This option is leveraged by the "Update Latest Date/Time on Scalar Measuring Component" algorithm type (D1-UPDDTSCMC) and the "Auto-Read Scalar Periodic Estimation" algorithm type (D1-ARSPE). It should be used in situations where Scalar Periodic Estimation is being implemented for devices that have an existing measurement history.

When implemented this option will instruct the "Update Latest Date/Time on Scalar Measuring Component" algorithm type (D1-UPDDTSCMC), which is responsible for maintaining the latest measurement date/time information for scalar measuring components, to skip initializing the latest measurement date/time information when an initial measurement is received for a measuring component where there is existing measurement data history. The initialization of the latest measurement date/time information should be deferred to the Auto-Read Scalar Period Estimation process. Note: this only applies for situations when the latest measurement date/time information has not been set previously and the initial measurement is not the first measurement received for the measuring component.

This option will instruct the "Auto-Read Scalar Period Estimation" process (algorithm type D1-ARSPE) to initialize the latest measurement date/time information on the measuring component, when the latest measurement date/time information is not set, to the most recent usable measurement that falls between the Process Date Time less this option value and the Process Date Time. For example, if this option is set to 744 (i.e. 31 days) and the Process Date Time is 02/01/2013 12:00:00AM, the algorithm would search for the most recent usable measurement between 01/01/2013 12:00:00AM (31 days prior to the Process Date Time) and 02/01/2013 12:00:00AM. The intent of this is to provide greater control over when Scalar Periodic Estimation will begin estimation for existing devices.

This feature configuration is only intended to be used for the first execution of the "Smart Meter State Monitor Process" batch control (D1-SMMTR), for subsequent executions the option should be removed.

Business Intelligence Configuration

Business intelligence configuration is used to define external data source indicators used when Oracle Utilities Meter Data Management is integrated with Oracle Utilities Business Intelligence. External data source indicators allow business intelligence extracts to properly link the external identifiers to the source external system. The Value of the Data Source Indicator option should match the Environment ID on the Installation Option of the external system.

Master Configurations

Master Configurations are sources of global parameter records used by a system implementation.

The Oracle Utilities meter data products use a Global Configuration record that controls core system functions. This record must be set up for the system to properly operate. See **Admin Setup Reference Tables** on page 3-10 for more information on when to set up this record.

Key concepts related to Master Configurations are discussed in this section. Refer to the embedded help for descriptions of the settings on the Master Configuration page.

Service and Measurement Data Foundation Master Configurations

The following master configurations are used by products that leverage the Oracle Utilities Service and Measurement Data Foundation, including Oracle Utilities Meter Data Management.

BI Master Configuration - ODI-Based

This master configuration is primarily used to configure the information that will be used in the extraction of data for business intelligence when using the Oracle Data Integrator extract, transform, and load (ETL) process.

BI Master Configuration - OWB-Based

This master configuration is primarily used to configure the information that will be used in the extraction of data for business intelligence when using the Oracle Warehouse Builder extract, transform, and load (ETL) process.

Generic BI Configuration

This master configuration defines options related to extraction of flat files used with Oracle Utilities Business Intelligence. Note that this master configuration is used by all products based on the Oracle Utilities Application Framework.

Hijri to Gregorian Date Mapping

The Hijri to Gregorian Date Mapping option is used to define the relationship between Hijri dates and Gregorian dates for each year.

ILM Configuration

This master configuration captures configuration details used by the Information Lifecycle Management (ILM) processes. Note that an instance of this master configuration is only required if your implementation uses information lifecycle management. This master configuration displays a list of the maintenance objects configured to support ILM as well the related eligibility algorithms and batch control crawlers for each.

Master Data Synchronization Configuration

The Master Data Synchronization Configuration option is used to define all foreign key references that need resolution. Each foreign key reference references the view that contains the external key / production key cross-reference. For entities that undergo both the initial and the ongoing synchronization, two views are specified. For entities that undergo the ongoing synchronization, an external system / ID type mapping is specified to cater for entities that might be synchronizing from more than one external system.

ODM Integration Master Configuration

The ODM Integration Master Configuration is used to define options related to integration of Oracle Utilities Operational Device Management to Oracle Utilities Meter Data Management. Specific options defined include the External System that represents the Oracle Utilities Operational Device Management application, its URL, and the number of hours Oracle Utilities

Meter Data Management waits for a response from Oracle Utilities Operational Device Management before transitioning an outbound synchronization request to the error state.

In addition, this configuration defines specific Maintenance Objects being synchronized and corresponding Outbound Message Types used to communicate its information to Oracle Utilities Operational Device Management.

Seeder Sync Master Configuration

The Seeder Sync Master Configuration is used to define the maintenance objects (device, device configuration, etc.) that require synchronization. Each maintenance object references the synchronization business object that needs to be instantiated when processing a synchronization request for that maintenance object. For maintenance objects that undergo both initial and the ongoing synchronization, two business objects are specified.

Service Order Management Master Configuration

This master configuration captures configuration details used in service order management processes, including appointment creation, exception handling, field activity cancellation conditions, restrictions related to cutting service for non-payment, and calculation of analytics used in the Service Order Management dashboards.

Smart Grid Gateway Master Configuration

This business object is used to globally enable and disable device event notification suppression. If this master configuration is added in Oracle Utilities Smart Grid Gateway and it is set to Enable notification suppression, device event notifications will be blocked when a notification suppression activity exists for the specific device event and its associated device or service point. If this master configuration is not added in the system or if it is set to Disable device event notification suppression, device event notifications will not be checked for suppression activities.

Meter Data Management Master Configurations

The following master configurations are used specifically with Oracle Utilities Meter Data Management.

MDM Master Configuration

The MDM Master Configuration is used when integrating Oracle Utilities Meter Data Management and Oracle Utilities Customer Care and Billing to define options related to calculating high and low boundary readings based on scalar readings stored in Oracle Utilities Meter Data Management that are used in billing in Oracle Utilities Customer Care and Billing.

Self Service Master Configuration

The Self Service Master Configuration defines options used when integrating Oracle Utilities Meter Data Management with Oracle Utilities Customer Self Service. Specific options defined include:

- **Temperature Source Factor** Applies to Get Usage Details service, when usage is overlaid with temperature information. Holds the temperature profile used to extract temperature information.
- **Temperature Mode** Applies to Get Usage Details service, when usage is overlaid with temperature information. Indicates the temperature profile's scale.
- **Usage Calculation Skip Option** Applies to Multiple Accounts Aggregation and Comparison service. Indicates how to proceed if one of the account's usage subscriptions is skipped. Specify 'All or Nothing' to skip the account if one of its usage subscriptions is skipped. Specify 'Allow Partial Usage' to continue with the summarization if a usage subscription is skipped.
- **Calculation Function** Applies to Multiple Accounts Aggregation and Comparison service. Indicates how the usage returned by the Get Usage Details service is summarized for the account. If not supplied, the usage is summarized based on the UOM's measures peak flag - 'Sum' if UOM does not measure peak or 'Max' if UOM measures peak.
- **Service Charges to Date Settings:** Used to define the acceptable window for displaying scalar data in Oracle Utilities Customer Self Service. The value of the "Daily Scalar Read Lag Window (Hours)" via a lag time parameter is subtracted from the request end date/time when retrieving data to display. If there is scalar data present through the request end date/time minus the lag time value, the data is returned for display.
- **Processing Scripts:** Used to retrieve information about measuring components, service points/device configurations, and usage subscriptions
- **Service Tasks:** Used to define self-service tasks available to Oracle Utilities Self Service users.
- **Rate Compare Configuration:** Used to configure how usage adjustments are applied when performing self-service rate comparisons. If your organization wishes to allow self-service customers to see the effect of usage adjustments on their consumption, configure the relevant usage adjustments possible for each rate defined in the CCB Rate Schedule extendable lookup. Types of usage adjustments (electric car, solar panels, etc.) are defined using the Usage Adjustment Type extendable lookup, and profile factors should be configured with measurement data representing the effect of each adjustment type. Customers can then choose one or more usage adjustments they would like to apply to their existing usage.
- **Supported Scalar Usage Groups:** Used to define the scalar usage groups supported when retrieving usage from Oracle Utilities Meter Data Management for access by Oracle Utilities Self Service users.
- **Usage Calculation Supported Usage Groups:** Used to define the usage groups supported when performing usage calculations initiated by Oracle Utilities Self Service users.

See the *Oracle Utilities Self Service Implementation Guide* for more information about configuring Oracle Utilities Meter Data Management for use with Oracle Utilities Self Service.

Timeliness Master Configuration

The Timeliness Master Configuration is used to define options that determine when an initial measurement is considered late and the severity its “lateness” (late vs very late vs very, very late, etc). These options is used in aggregation of measurement data. Specific options defined include factors used when calculating heating and cooling degree days, the number of hours after which an initial measurement is considered late, and a set of buckets that specify degrees of “lateness” for each of the Value Identifier Types used by the aggregator measuring components. For example, measurements that are “On Time” might be placed in “Value 1”, while measurements that are between 5 and 10 hours “late” might be placed in “Value 2”, and measurements that are between 10 and 15 hours late might be placed in “Value 3”.

Usage Transaction Export Configuration

The Usage Transaction Export Configuration is used to configure properties that control the export of usage transaction data. It holds the device and service point business objects used to read those respective objects to supply information for use by a customer information system's bill print system. It also defines the estimation threshold percentage above which a service quantity being exported via the usage transaction outbound message is marked as an estimate.

Chapter 3

Setting Up Admin Data

This chapter describes the different types of admin data that must be set up and defined as part of implementing and configuring Oracle Utilities Meter Data Management, including:

- **Understanding Admin Data**
- **Admin Setup Reference Tables**

Understanding Admin Data

This section describes the admin data used by the Oracle Utilities Service and Measurement Data Foundation and Oracle Utilities Meter Data Management.

General Admin Data

General admin data are types of data used by multiple functional areas.

Exception Types

Exception types define the properties common to many exceptions.

When creating validation, editing, and estimation (VEE) rules, you might create an exception type for each VEE rule. You might also create more general exception types, such as "Insufficient Data" to be used to signify that a measurement didn't have sufficient data for the VEE rule to execute.

Factors

Factor are a centrally stored set of values for use in validation rules, bill determinants calculations, and other processes.

A factor can have different values depending upon some definable attribute of a system object, such as customer size associated with a service point. Examples of factors can include minimum/maximum thresholds, loss factors, etc. Classes of factors are defined that can have numeric values (as in the above examples), or values pointing to profile measuring components, or VEE groups.

A factor's values are effective-dated values - either a number, a profile measuring component, a VEE group, or some custom-defined value - assigned to a factor and associated to the value of some attribute of a system object. For example, consider a service point that can be classified as residential, commercial, or industrial. The tolerance percentage by which a customer's consumption can exceed last month's consumption can be based on the service point category. For this example, factor values for a single factor called "tolerance percentage" could be:
Residential - 20% Commercial - 10% Industrial - 5%.

Service Providers

Service providers are external entities that serve various roles relative to the application.

Service providers can include head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that receive meter event data from the application, or other parties that require or provide information to the system.

Service Quantity Identifiers

Service Quantity Identifiers (SQI) are used to further distinguish between measured quantities that have identical UOM/TOU combinations, including situations in which the distinguishing identifier of a UOM is not accurately described as a TOU.

SQIs can also be used as a stand-alone representation of a service quantity that is not measured (one that is not properly described as a UOM) within a usage service quantity collection (such as a billing determinant).

Service Types

Service Types define specific types of service for which usage can be recorded and captured, such as electric, gas, steam, etc.

Time of Use

Time of Use (TOU) periods are modifiers for a given unit of measure that indicate a period of time during which a quantity has been used, such as On-Peak (meaning during a time when the greatest quantity of some consumable is being used), Off-Peak (meaning during a time when the least amount of some consumable is being used), etc.

Units of Measure

Units of Measure (UOM) identify quantities measured and recorded, such as KWH, KW, cubic feet, degrees Celsius, etc. UOMs are based on a specific service type.

Attributes used to define units of measure include the following:

- **Service Type:** The type of service (electric, gas, etc.) measured by the UOM
- **Decimal Positions:** The number of decimal places used when sending usage transaction service quantities for this UOM to Oracle Utilities Customer Care and Billing by way of the base package Usage Transaction Outbound Message
- **Allowed on Measuring Component:** A flag that indicates if the UOM is allowed on Measuring Components
- **Measures Peak Quantity:** A flag that indicates if the UOM is used to measure peak quantities or not. An example of a UOM that measures peak quantities is kilowatts (KW).
- **Magnitude:** A number that indicates the relative size of the UOM as compared to a single unit of the UOM specified under "Base Unit of Measure." For example, megawatt hours (MWH) have a magnitude of 1,000 as compared to a single kilowatt hour (KWH).
- **Base Unit of Measure:** The UOM upon which the current UOM is based. Used in conjunction with magnitude. For example, the base unit of measure for megawatt hours (MWH) with a magnitude of 1,000 would be kilowatt hours (KWH).

Device Management Admin Data

Device management admin data include data that defines “types” of device-related objects.

Device Configuration Types

Device configuration types define the properties of device configurations of a given type, including the valid types of measuring components that can be configured for device using configurations of this type.

Item Configuration Types

Device configuration types for items are created using the "Item Configuration Type" business object (D1-ItemConfigurationType).

Device Types

Device types define information about a class of devices, including properties that apply to all devices of a type. Properties defined for a device type can be overridden for an individual device.

Item Types

Device types for items are created using the “Item Type” business object (D1-ItemType).

Manufacturers

Manufacturers are the companies that makes devices. A device's manufacturer is defined as an attribute of the device itself.

Each manufacturer can have zero or more models defined. Models for a single manufacturer can have diverse service types.

Measuring Component Types

Measuring component types define the most important properties of a measuring component.

Measuring component types define what a measuring component measures (KWH, temperature, etc.), how regularly it measure it, and whether it should be connected to a physical device, or if it's used as a scratchpad measuring component or an aggregator measuring component. Measuring component types also specify how the measuring component's final measurements should be stored, how the measuring component's user-defined values should be calculated, and specific rules governing validation, editing, and estimation (VEE) for measuring components of the type. In addition, measuring component types define display properties and valid attribute values for measuring components belonging to the type.

Some important characteristics defined for measuring component types include:

- **Value Identifiers:** These store the values of UOM, TOU, and SQI that identify the measured amounts for measuring components of this type. Value identifiers specify the quantities stored on the measurement records for measuring components of this type.
- **Valid VEE Groups:** These define the VEE groups considered valid for measuring components of this type.
- **Fallback VEE Groups:** These define default VEE groups that can be used with all measuring components of this type. This alleviates the need to specify the same VEE groups on multiple measuring components of the same type. Each VEE group is designated a VEE group role that indicates when and how the VEE group is used (for initial load, manual override, or estimation).
- **Eligible Profile Factors (interval only):** These define the profile factors that are considered to be eligible for interval measuring components of this type. You can also specify one or more profile factors as a default.
- **Valid Profile Factors for Conversion from Scalar to Interval (scalar only):** These define the profile factors that are considered to be eligible for scalar measuring components of this type when converting scalar measurements to interval measurements. You can also specify one or more profile factors as a default.
- **Valid Scratchpad Measuring Component Types:** These define the scratchpad measuring component types considered valid for measuring components of this type.
- **Display Properties:** Defines how measurement data for measuring components of this type is displayed, including:
 - **Display Configuration:** Details related to how measurements are displayed, including the number of hours of data to display, the default TOU map used, the TOU by Day Profile factor used, and default measurement condition.
 - **Event Bar Profiles:** The event bar profiles used when displaying measurement data for measuring components of this type. Event bar profiles are defined as values for the 360 View Event Bar Profile extendable lookup.
 - **Final Values Overlay Profiles:** The final values overlay profiles used when displaying measurement data for measuring components of this type. Final values overlay profiles are defined as values for the Final Values Overlay Profile extendable lookup.

Measuring component types are described in more detail in **Chapter 4: Devices, Measuring Components, and, Device Configurations**.

Device Installation Admin Data

Device installation admin data includes data used to support the installation of devices.

Markets

Markets define the jurisdictions or regulatory environments in which a service point participates.

Markets also define market relationships for valid service providers and their roles within a market (distributor, etc.). While each service point specifies only one market, a utility may serve more than one market, and different service points throughout the utility's service territory can be linked to different markets.

Service Point Types

Service point types define a specific type of point at which service is delivered.

Specifically, service point types define how the application manages many aspects of the service point's behavior. A service point type may have one or more valid device types defined that limit the types of devices that can be installed at service points of this type.

Service Point Category

The "Service Point Category" field defines the types of devices that can be installed at service points of this type. Valid values include:

- **Meter:** Indicates that a single meter can be installed at service points of this type.
- **Item:** Indicates that a single "badged" item can be installed at service points of this type.
- **Multi-Item:** Indicates that one or more "unbadged" items can be installed at service points of this type.

Contact Types

Contact types define the properties of a class of entities (businesses, persons).

Measurement Cycles

Measurement cycles define the schedule for manual meter reading of devices at service points in that cycle. Measurement cycles can have one or more associated routes used to collect measurements.

When used with smart meters, measurement cycles can also be configured to define when to create usage transactions for usage subscriptions associated to service points in the cycle.

Measurement Cycle Schedules

Measurement cycle schedules define the dates on which devices are scheduled to be read for a given measurement cycle and the routes used to collect measurements for the measurement cycle.

Device Communication Admin Data

Device communication admin data includes data used to support communication with head-end systems.

Activity Types

Activity types define properties common to a specific type of activity.

Activity types include types of communications between an application and a head-end system, such as a connection requests, meter ping requests, or on-demand meter readings, as well as device event types.

Communication Types

Communication types define properties common to a specific type of communication.

Communication types include types of communications between an application and a head-end system, such as notifications (used to notify an head-end system of a command request), or message responses (sent from a head-end system to confirm receipt of a message).

Note: Communication types are applicable only with Oracle Utilities Smart Grid Gateway.

Device Event Types

Device event types define properties common to specific types of events.

Device event types represent different types of events that can take place relative to a device. Examples of device events include power outages, power restoration, tampering alerts, and other events.

Device event types can be defined by the following attributes:

- **Standard Event Name:** the "standard" name of the event type in Smart Grid Gateway. Device vendors may have their own specific names for device events.
- **Device Event Category:** a category (defined as an Extendable Lookup) used to group device event types.
- **Reporting Category:** a category used to group device event types for reporting purposes.
- **Activity Type:** the activity type for activities created for device events of this type.

Service Task Types

Service tasks types define properties common to specific types of service tasks.

Service task types represent types of tasks that can be performed by users of other Oracle Utilities applications, such Oracle Utilities Customer Self Service or Oracle Utilities Network Management System. Examples of service tasks include self service meter reads, in which users enter their own meter reads via the Customer Self Service application, and service issue monitor types used when determining if service investigation is needed for a service point.

VEE Rule Admin Data

VEE rule admin data include VEE groups and VEE rules.

VEE Groups

VEE groups are collections of VEE rules that are applied to initial measurement data.

VEE groups can be associated to a specific measuring component, or to a measuring component type (or both). VEE groups associated with a measuring component type are applied to all measuring components of that type, while those associated to a specific measuring component are applied only to that measuring component.

VEE Rules

VEE rules are standard and custom Validation, Estimation and Editing (VEE) rules that perform checking and/or manipulation of initial measurement data.

VEE rules are created for a specific VEE group. For example, if you were configuring two VEE groups and both included a specific VEE rule, you would need to create two instances of the VEE rule, one for each group.

Attributes used to define VEE rules typically include the following:

- **Basic Information:** Basic information about the VEE rule, including its name and description, the VEE group to which the rule belongs, the sequence of the rule within the

group, the category, and start and end dates. This information is standard for most VEE rules.

- **Parameters:** The parameters used by the rule. Parameters are specific to each rule.
- **Exception Types and Severity:** Details about how to handle exceptions, including the Exception Type and Exception Severity for exceptions created by the rule.

VEE Rule Eligibility Criteria

VEE rule eligibility criteria are user-definable conditions that could cause a given VEE rule to be applied or skipped. This can involve the evaluation of some attribute of the device or measuring component, or something else entirely.

A VEE rule can have multiple eligibility criteria for determining if the rule should be applied or skipped, based on a user-defined sequence.

Usage Management Admin Data

Usage management admin data includes data used in usage calculations, including time of use data, usage subscription types, and usage groups and rules. Usage subscription types are described in more detail in **Chapter 9: Usage Subscriptions**. Usage groups and rules are described in more detail in **Chapter 10: Usage Groups and Usage Rules**.

Dynamic Option Types

Dynamic option types store information common to dynamic options of a specific type.

TOU Groups

TOU Groups are groups of TOUs used to limit the set of TOUs usable in a TOU schedule. TOU groups are used when defining a TOU schedule via a TOU map template.

TOU Map Templates

TOU Map Templates are the schedules used for TOU map data generation.

Attributes used to define TOU map templates include the following:

- **TOU Group:** the TOU group used by the map template
- **Default TOU:** the default TOU for the map template (from the TOU Group). This is the TOU used when creating TOU map data for dates not accounted for in the TOU Schedules section.
- **Work Calendar:** the work calendar associated with the map template. Work calendars define the days of the week on which work is performed, and specify holidays.
- **Holiday TOU:** the TOU used for holidays (from the TOU Group)
- **Holiday Template:** the TOU map template used for holidays (if applicable)
- **Interval Size:** the size of the intervals for TOU map data created from the map template, represented as hours:minutes:seconds (HH:MI:SS).
- **TOU Schedules:** date ranges (including month, day, and time ranges) and which TOUs should be used during each.

TOU Map Types

TOU Map Types define important properties of TOU maps of the type, including the interval size and the valid TOU map templates.

Attributes used to define TOU map types include the following:

- **Time Zone:** the time zone in which TOU maps of this type are applicable
- **Interval Size:** the size of the intervals for TOU map data created from maps of this type, represented as hours:minutes:seconds (HH:MI:SS).
- **Default TOU Map Template:** the default TOU map template used by maps of this type
- **Override TOU Map Templates:** one or more TOU map templates that can be used as an override on TOU maps of this type.

Usage Subscription Types

Usage Subscription Types define a collection of properties defining a class of usage subscriptions. Usage subscription types also control valid values for various attributes of usage subscriptions.

Attributes used to define usage subscription types include the following:

- **Service Provider:** The service provider for usage subscriptions of this type

- **Valid Service Point Types:** One or more service point types considered valid for usage subscriptions of this type
- **Valid Service Providers:** One or more service providers considered valid for usage subscriptions of this type
- **Valid Usage Groups:** One or more usage groups considered valid for usage subscriptions of this type
- **Fallback Usage Groups:** One or more fallback usage groups for usage subscriptions of this type. Fallback usage groups are used in the event that a usage group defined for a usage subscription is not in effect at the time usage is to be calculated.

Usage Groups

Usage groups are collections of usage rules that are applied to measurement data to calculate bill determinants for usage subscriptions.

Usage groups are associated with specific usage subscriptions and usage subscription types (or both). When assigned to usage subscriptions, usage groups contain the usage rules to be used to calculate usage and bill determinants. Usage groups associated with usage subscription types are those groups considered valid for usage subscriptions of that type.

Usage groups can also specify a list of device configuration types that are considered valid. Usage groups should only be associated with usage subscriptions for service points related to device configurations of a valid device configuration type.

Usage Rules

Usage rules are standard and custom rules that perform calculations on measurement data to generate bill determinants and other values used by external systems, such as billing systems, customer information systems, etc.

Usage rules are created for a specific usage group. For example, if you were configuring two usage groups and both included a specific usage rule, you would need to create two instances of the usage rule, one for each group.

Attributes used to define usage rules typically include the following:

- **Basic Information:** Basic information about the usage rule, including its name and description, the usage group to which the rule belongs, the sequence of the rule within the group, and the usage rule category. This information is standard for most usage rules.
- **Parameters:** The parameters used by the rule. Parameters are specific to each rule.

Usage Rule Eligibility Criteria

Usage rule eligibility criteria are user-definable conditions that could cause a given usage rule to be applied or skipped. This can involve the evaluation of some attribute of the usage subscription or service point, or something else entirely.

A usage rule can have multiple eligibility criteria for determining if the rule should be applied or skipped, based on a user-defined sequence.

Admin Setup Reference Tables

This section lists and describes all objects that must be defined as part of the setup process for Oracle Utilities Meter Data Management. It identifies the order in which objects should be defined and any prerequisites for setup.

Note: All basic Framework setup, including system and database setup and any modifications or extensions to base business objects, must have been completed before beginning setup tasks for Oracle Utilities Meter Data Management. See the Framework documentation for more information.

Setup Sequence

In the setup tables that follow, the **Sequence** column displays the following codes:

L1 = Object has no setup prerequisites and should be defined before L2-L6 objects.

L2 = Object has some L1 prerequisites and should be defined after all L1 objects have been defined and before L3 objects.

L3 = Object should be defined after all L1 and L2 objects have been defined.

L4 = Object should be defined after all L1, L2, and L3 objects have been defined.

L5 = Object should be defined after all L1, L2, L3, and L4 objects have been defined.

Administration Setup and Maintenance

To access the maintenance portals for the objects in this section, do one of the following:

- If you are using functional menus, select **Admin Menu**>[*Functional Menu*]>[*object name*]
- If you are using alphabetical menus, select Admin Menu>[*object name*]

The [*Functional Menu*] and [*object name*] are provided in the appropriate columns in the following tables.

Application Framework Setup

Seq	Object	Functional Menu	Description	Prerequisites
L1	Country	General	Your organization's country.	None
L1	Currency	Financial	Your organization's native currency.	None
L1	Display Profile	General	Controls how dates, times, and numbers displayed.	None
L1	Language	General	The language to use for this implementation.	None
L1	Time Zone	General	Your organization's base time zone.	None
L1	To Do Role	General	Used to associate users with To Do entries.	None
L1	Work Calendar	General	The work calendar for your organization, which identifies your public holidays	None
L2	Installation Options	System	Control various aspects of the system. Refer to the Installation Options section earlier in this document.	Time Zone, Language, Currency

Seq	Object	Functional Menu	Description	Prerequisites
L2	Master Configuration	System	Enables an implementation to capture various types of information in the system.	
L2	To Do Type	General	Used to define types of To Do Entries	To Do Role
L2	User	Security	Defines a user's user groups, data access roles, portal preferences, default values, and To Do roles	Language, Display Profile, To Do Roles
L2	User Group	Security	A group of users who have the same degree of security access	User

Oracle Utilities Meter Data Management Setup

Seq	Object	Functional Menu	Description	Prerequisites
L1	Activity Type	Communications	Defines properties common to a specific type of activity	None
L1	Communication Type	Communications	Define properties common to a specific type of communication	None
L1	Contact Type	Customer Information	Defines properties of a class of entities (businesses, persons)	None
L1	Device Event Type	Communications	Defines properties common to specific types of events	None
L1	Exception Type	Common	Defines properties common to exceptions of a specific type	None
L1	Factor	Common	Centrally stored sets of values for use in validation rules, bill determinants calculations, and other processes	None
L1	Market	Communications	Defines jurisdictions or regulatory environments in which a service point participates	None
L1	Measurement Cycle	Device Installation	Defines the schedule for manual meter reading of devices at service points in that cycle	None
L1	Measurement Cycle Schedule	Device Installation	Define the dates on which devices are scheduled to be read for a given measurement cycle	None
L1	Service Provider	Communications	External entities that serve various roles relative to the application (head-end systems, billing systems, market participants, outage management systems, etc.)	None
L1	Service Quantity Identifier	Common	Used to further distinguish between measured quantities that have identical UOM/TOU combinations	None

Seq	Object	Functional Menu	Description	Prerequisites
L1	Service Task Type	Communications	Defines specific types of tasks performed by external users (self-service meter reads, self-service outage notifications, etc.)	None
L1	Service Type	Common	Defines specific types of service for which usage can be recorded and captured (electric, gas, steam, etc.)	None
L1	Time of Use	Common	Modifiers for a given unit of measure that indicate a period of time during which a quantity has been used (On-Peak, Off-Peak, etc.)	None
L1	VEE Group	VEE Rules	Collections of VEE rules that are applied to initial measurement data	None
L2	Dynamic Option Type	Usage	Defines information common to dynamic options of a specific type	Time Zone
L2	Manufacturer	Device	Individual companies that makes devices. Manufacturers also reference models.	Service Type
L2	TOU Group 2	Usage	Groups of TOUs used to limit the set of TOUs usable in a TOU schedule	Time of Use
L2	Unit of Measure	Common	Quantities measured and recorded by the system (CCF, KWH, KW, etc.)	Service Type
L2	VEE Rule	VEE Rules	Standard and custom VEE rules that perform checking and/or manipulation of initial measurement data	VEE Group, Exception Type
L3	Measuring Component Type*	Device	Defines the most important properties of a measuring component	Factor (Profile), Service Quantity Identifier, Service Type, Time of Use, Unit of Measure, VEE Group
L3	TOU Map Template	Usage	Schedules used for TOU map data generation	TOU Group, Work Calendar
L4	Device Configuration Type	Device	Defines properties of device configurations of a given type	Service Type, Measuring Component Type
L4	Device Type	Device	Defines information about a class of devices	Service Type, Device Configuration Type

Seq	Object	Functional Menu	Description	Prerequisites
L4	TOU Map Type	Usage	Define important properties of TOU maps of the type	Time Zone, TOU Map Template
L4	Usage Group	Usage Rules	Collections of usage rules that are applied to measurement data to calculate bill determinants for usage subscriptions	Device Configuration Type
L5	Service Point Type	Device Installation	Defines specific types of points at which service is delivered	Service Type, Device Type
L5	Usage Rule	Usage Rules	Standard and custom rules that perform calculations on measurement data to generate bill determinants and other values used by external systems	Service Quantity Identifier, Time of Use, Unit of Measure, Usage Group
L5	Usage Subscription Type	Usage	Defines collections of properties defining a class of usage subscriptions	Service Point Type, Service Provider, Usage Group

* Measuring component types also reference other measuring component types, TOU maps, and extendable lookups.

Chapter 4

Devices, Measuring Components, and, Device Configurations

This chapter provides descriptions of devices, device configurations, and measuring components, including:

- Understanding Devices, Measuring Components, and Device Configurations
- Devices In Detail
- Device Configurations In Detail
- Measuring Components In Detail
- Measuring Component Types In Detail
- Configuring Devices, Measuring Components, and Device Configurations

Understanding Devices, Measuring Components, and Device Configurations

This section provides an overview of devices, measuring components, and device configurations, including how they are used in the Service and Measurement Data Foundation and related products including Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway.

Devices

Devices are physical or virtual objects that hold one or more measuring components that can produce data to be handled by the system. While most devices are meters, an implementation might set up devices for every asset that measures or monitors resource usage. For example, a device could be set up to record average daily temperature (if temperature plays a part in usage calculations). Examples of devices include meters, substations, transformers, demand response devices, weather stations, etc.

Device Categories

A device's "Device Category" (defined as an option on the device's business object) defines the broad category to which a particular device belongs. This option is used by smart meter commands and other processes when determining how the system should behave when executing processes based on the category of device. For example, when viewing items via the Device portal, certain item-specific zones appear that are not displayed when viewing meters or communication components. The base package supports the following device categories (defined as values for the DEVICE_CATEGORY_FLG lookup):

- Smart Meter (D1SM)
- AMR Meter (D1AR)
- Manual Meter (D1MN)
- Item (D1IT)
- Communication Component (D1CC)

Items

While the most common types of devices are meters, there are many other devices that can be involved with a customer's service. These other devices are referred to simply as "items", and are used for many different devices including lamps, poles, current transformers, backflow devices, pulse initiators, etc.

Items can be either "badged" or "unbadged". "Badged" items are those that have unique identifiers, and are represented by device records. Device records are not created for "unbadged" items. "Badged" items are installed at service points in much the same way as meters are installed (though they use an item-specific install event). For "unbadged" items, "multi-item" service points are used to define the number of each TYPE of item that is installed at the service point.

Communication Components

Communication components are devices attached to meters. A communication component may handle electronic transmission of measurement data, events, and commands in situations in which the underlying meter is not capable or not enabled to handle this data. Devices of this sort are sometimes referred to as ERT (Electronic Receiver/Transmitter) meters, or communication modules (for example, the term "gas module" may refer to a communication module attached to a gas meter).

Measuring Components

Measuring components are single points for which data will be received and stored in the system.

Types of Measuring Components

A measuring component can be associated to a physical device, which can have one or more measuring components, or it can be "virtual" or "stand-alone," meaning that it is not associated to a physical device. The Service and Measurement Data Foundation supports the following types of measuring components:

- **Physical:** Physical measuring components are those that physically exist, and that are linked to a device that can be configured differently over time. Interval channels and scalar registers are examples of physical measuring components.
- **Note:** The terms register and channel are synonyms for measuring component.
- **Standalone:** A standalone measuring component is used to record measurements for something that does not have a physical presence. For example, you might create a standalone measuring component to record the average daily temperature supplied by a weather station.
- **Scratchpad:** User create scratchpad measuring components to experiment with measurement manipulation functions before applying the functions to a physical or standalone measuring component. Examples of measurement manipulation might include experimenting with the impact of executing the "spike smooth" function on an initial measurement, or adding or removing intervals to or from the measurement. Scratchpad measuring components provide users with a means to manipulate "scratchpad" measurement data without affecting existing "live" measurement data.
- **Aggregator:** An aggregator measuring component holds summarized usage from other measuring components. For example, aggregator measuring components could be configured to hold total consumption for each postal code within a service territory.

Head-end system processing statistics used by Oracle Utilities Smart Grid Gateway are stored as aggregated measurements for aggregator measuring components.

Scalar vs. Interval

Beyond the four types described above, measuring components generally fall into one of two primary classes of: scalar measuring components, and interval measuring components.

- **Scalar** measuring components are measured at unpredictable intervals. For example, "once-a-month" is not a predictable interval as the amount of time between reads is unpredictable and inconsistent.

Scalar measuring components are typically read manually

- **Interval** measuring components are measured at predictable intervals, such as every 15 minutes, every 30 minutes, every hour, etc.

The terms "interval size" and "seconds-per-interval" (SPI) are used to define the size of an interval measuring component's intervals.

Note: A device may have any combination of interval and/or scalar measuring components

Measuring Component Measurements

Measuring components are configured to measure specific types of quantities. These include:

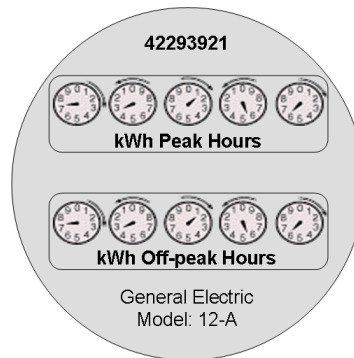
- **Unit of Measure:** The unit of measure for the quantity being recorded. Examples include kilo-watt hours (kWh), kilo-watts (kW), therms, cubic feet (CCF), temperature (Fahrenheit or Celsius), etc.
- **Time of Use:** Modifiers for a given unit of measure that indicate a period of time during which a quantity has been used, such as On-Peak (meaning during a time when the greatest

quantity of some consumable is being used), Off-Peak (meaning during a time when the least amount of some consumable is being used), etc.

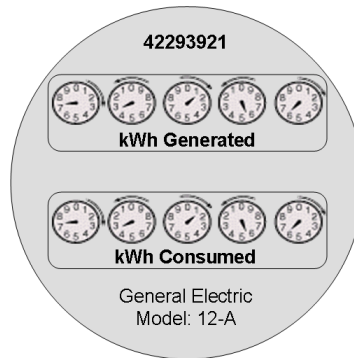
- **Service Quantity Identifiers:** Used to further distinguish between measured quantities that have identical UOM/TOU combinations, including situations in which the distinguishing identifier of a UOM is not accurately described as a TOU. Generally, SQI is only used when multiple measuring components measure the same thing, but in different ways. A meter that measures both generation KWH and consumption KWH could use SQIs to differentiate between the two.

The combination of UOM, TOU and SQI define what a measuring component measures. TOU and SQI are optional, but UOM must be defined for all measuring components.

For example, consider a meter (as illustrated in the image below) with two measuring components, both measuring the same unit of measure (kWh), but each measuring component measures consumption in different time of use (TOU) periods (peak and off-peak).



Another example might be a meter that records both generated KWH and consumed KWH. This meter would be configured to measure both UOM and SQI.



A measurement is recorded each time a measuring component is measured. This means that for a meter with two measuring components that is read once a month, two measurements, one for each measuring component, would be recorded each month.

Subtractive vs. Consumptive Measurements

Another attribute that defines how measuring components measure quantities is the distinction between subtractive and consumptive measuring components.

A subtractive measuring component's usage is equal to the current measurement (also known as the Stop or End Measurement or Reading) minus the previous measurement (also known as the Start Measurement or Reading). To put this more simply:

$$\text{Usage} = \text{End Measurement} - \text{Start Measurement}$$

Most residential scalar KWH meters are subtractive. The table below lists a series of measurements for a subtractive measuring component.

Date	KWH (Measurement)	Usage
01/01/2010	0	
01/31/2010	1000	1000 KWH
03/02/2010	1789	789 KWH
04/01/2010	2700	911 KWH

A consumptive measuring component's usage is equal to its current measurement. Consumptive measuring components are often used to measure demand, such as KW. The table below lists a series of measurements for a consumptive measuring component.

Date	KW (Measurement)	Usage
01/01/2010	0	
01/31/2010	10	10 KW
03/02/2010	6	6 KW
04/01/2010	5	5 KW

Interval measuring components can also be considered consumptive, in that the consumption value of each individual interval is equal to its measurement.

Interval vs. Scalar Measurements

As noted above, interval measuring components record measurements every interval, defined by the measuring component's interval size (measuring in seconds per interval). For interval measuring components, measurements are only allowed on these time boundaries. For example, measurements for an interval measuring component with an interval size of 15 minutes (or SPI of 900) on January 1, 2010 might be as follows:

Date	Time	KWH
01/01/2010	10:00 AM	0
01/01/2010	10:15 AM	10
01/01/2010	10:30 AM	6
01/01/2010	10:45 AM	5

In contrast to interval measuring components, scalar measuring components are read at unpredictable (and often inconsistent) intervals and are allowed at any point in time. In practice, scalar measuring components are read monthly, bimonthly, quarterly, etc. For example, measurements for a scalar measuring component from January 1, 2010 through April 1, 2010 might be as follows

Date	KWH (Measurement)
01/01/2010	0
01/31/2010	1000

Date	KWH (Measurement)
03/02/2010	1789
04/01/2010	2700

Note that interval and scalar measuring components can exist on the single meter. For these types of meters, the scalar measuring component is typically used to verify and validate the interval measurements. For example, the sum of all the interval measurements within a measurement period should equal the scalar measurement for the same period.

Device vs. Measuring Component Attributes

The distinction between attributes used to define devices and measuring components is important when creating devices and measuring components as part of an implementation. For example, if you identify additional attributes you wish to capture, it's important to store those attributes in the most appropriate place.

Devices have attributes that are applicable to the physical object, and that are the same regardless of the number of measuring components on the device. For example:

- The type of device
- Manufacturer and Model
- Serial Number
- Badge Number
- Head-End System (for smart meters)

Measuring components have attributes that may differ for each measuring component on a device, for example:

- The type of measuring component (which in turn defines the measuring component's UOM, TOU, SQI, whether it is scalar or interval (and its interval size), and others)
- Channel ID (for interval channel measuring components)
- Channel (or Register) multiplier (a value by which the measured consumption is multiplied to derive usage)
- Validation, Editing, and Estimation groups used when validating initial measurement data for the measuring component.

Device Configurations

A measuring component's attributes can change over time. Device configurations record how a device's measuring components look at an instant in time. A new device configuration is required whenever a device's measuring components are reconfigured. For example, if the register multiplier on a measuring component changes as of June 1, 2010, the device would require a new device configurations dated 1-Jun-2010 to reflect the change.

Note that device configurations don't typically capture the changed information, but instead indicate that changes of some sort have taken place on one or more of the device's measuring components.

Items and Device Configurations

"Badged" items use device configurations in much the same way as those used for meters and measuring components. Item-based device configurations are based on the "Item Configuration Type (D1-ItemConfigurationType) business object and can be installed at service points (the "Item Install Event" business object is used for install events for "badged" items).

Communication Components and Device Configurations

Communication components can have device configurations (one or more over time) with measuring components to record initial measurements, and its device configuration can be installed at a service point (the "Communication Component Install Event business object is used for install events for components).

Devices In Detail

This section provides details concerning the device objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom devices you create as part of your implementation. This section includes:

- A description of the D1-DEVICE maintenance object
- Lists of the base package device business objects, including “lite” business objects
- Details concerning device-specific configuration options
- A sample device business object (D1-SmartMeter)

Maintenance Object - D1-DEVICE

Device business objects use the D1-DEVICE maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-DEVICE
Description	Device
Service Name	D1-DEVICE (Device Maintenance)
Tables	<ul style="list-style-type: none">• D1_DVC (Device) - Primary• D1_DVC_CHAR (Device Characteristics) - Child• D1_DVC_IDENTIFIER (Device Identifier) - Child• D1_DVC_LOG (Device Log) - Child• D1_DVC_LOG_PARM (Device Log Parameter) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Device Business Objects

The Service and Measurement Data Foundation base package includes the following device business objects:

Business Object Name	Description
D1-CommComponentDevice	Communication Component Device Instances of this business object represent individual communication components defined in the system.
D1-Item	Item Instances of this business object represent individual items defined in the system.
D1-ManualMeter	Manual Meter Instances of this business object represent individual manual meters defined in the system.

Business Object Name	Description
D1-SmartMeter	Smart Meter Instances of this business object represent individual smart meters defined in the system.

The Service and Measurement Data Foundation base package includes the following “lite” device business objects:

Business Object Name	Description
D1-DeviceDetailsLITE	Device LITE
D1-DeviceIDsLire	Lite BO to Get AMI ID Type dynamically
D1-DeviceIdentifierLite	Device Identifiers LITE
D1-DeviceLite	Device LITE
D1-DeviceLiteAMI	BO to Get AMI related details for Deive
D1-DeviceParentLite	Device Parent LITE
D1-ItemLITE	Item LITE

The Service and Measurement Data Foundation base package includes the following additional device business objects:

Business Object Name	Description
D1-SynchronizationAddDevice	Device Synchronization Add (used when adding a new device as a result of a data synchronization request)

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by device business objects.

Business Object Options

Device business objects can make use of the following business object options:

- **Device Category:** This option defines the category to which the device belongs (smart meter, AMR meter, manual meter, item, or communication component). This option is used by smart meter commands and other processes based on the general type of device.
- **Install Event BO:** This option identifies the install event business object to use when installing device configurations for devices defined by this business object. For example, for the D1-SmartMeter device, this option is set to D1-SmartMeterInstallEvent, meaning that any time a device configuration for the D1-SmartMeter device is installed, the install event business object used will be D1-SmartMeterInstallEvent.
- **Synchronization Add BO:** This option identifies the business object to use when adding new devices as a result of a data synchronization request.
- **Valid Command Request BO:** This option defines the valid commands available for the device, and the activity business object to use for each command. This option is used with Oracle Utilities Smart Grid Gateway and can be defined multiple times for the same device, once for each command supported by the device.

For example, for the D1-SmartMeter device, this option is defined seven times, once for each of the following commands: On-Demand Read - Interval, On-Demand Read - Scalar, Device Status Check, Remove Connect, Remote Disconnect, Device Commission, and Device Decommission.

Example Device - D1-SmartMeter

This section lists some of the details of the D1-SmartMeter device business object.

Option/Field	Description
Business Object	D1-SmartMeter
Description	Smart Meter
Maintenance Object	D1-DEVICE (Device)
Application Service	D1-SMARTMTRBOAS (Smart Meter BO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> • Device Category: D1SM (Smart) • Install Event BO: D1-SmartMeterInstallEvent (Smart Meter Installation Event) • Synchronization Add BO: D1-SynchronizationAddBO (Device Synchronization Add) • Valid Command Request BOs: <ul style="list-style-type: none"> • D1-OnDemandReadInterval (On-Demand Read Interval) • D1-OnDemandReadScalar (On-Demand Read Scalar) • D1-DeviceStatusCheck (Device Status Check) • D1-RemoteConnect (Remote Connect) • D1-RemoteDisconnect (Remote Disconnect) • D1-DeviceCommission (Device Commission) • D1-DeviceDecommission (Device Decommission) • Display UI Map: D1-SmartMeterDisplay (Smart Meter - Display) • Portal Navigation Option: d1dvcTabMenu (Device) • Display Map Service Script: D1-RtSmMtrDs (Smart Meter - Retrieve Details for Display) • Maintenance UI Map: D1-SmartMeterMaint (Smart Meter - Maintenance)
Algorithms	<ul style="list-style-type: none"> • Information: D1-SMARTINFO (Smart Meter Information) • Validation: D1-VALRETDT (Validate Device Retirement Date)
Lifecycle	<ul style="list-style-type: none"> • Active (Initial) • Retired (Final)

Use the Business Object portal to view additional details concerning this business object.

Device Configurations In Detail

This section provides details concerning the device configurations supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom device configurations you create as part of your implementation. This section includes:

- A description of the D1-DVCCONFIG maintenance object
- Lists of the base package device configuration business objects, including “lite” business objects
- A sample device configuration business object (D1-DeviceConfiguration)

Maintenance Object - D1-DVCCONFIG

Device configuration business objects use the D1-DVCCONFIG maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-DVCCONFIG
Description	Device Configuration
Service Name	D1-DVCCONFIG (Device Configuration Maintenance)
Tables	<ul style="list-style-type: none"> • D1_DVC_CFG (Device Configuration) - Primary • D1_DVC_CFG_CHAR (Device Configuration Characteristics - Child • D1_DVC_CFG_LOG (Device Configuration Log) - Child • D1_DVC_CFG_LOG_PARM (Device Configuration Log Parameter) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Device Configuration Business Objects

The Service and Measurement Data Foundation base package includes the following device configuration business objects:

Business Object Name	Description
D1-DeviceConfiguration	Device Configuration Instances of the business object represent individual device configurations defined in the system.

The Service and Measurement Data Foundation base package includes the following “lite” device configuration business objects:

Business Object Name	Description
D1-DeviceConfigParentLITE	Device Configuration Parent LITE
D1-DeviceConfigurationLite	Device Configuration LITE

The Service and Measurement Data Foundation base package includes the following additional device business objects:

Business Object Name	Description
D1-SynchronizationAddDC	Device Configuration Synchronization Add (used when adding a new device configuration as a result of a data synchronization request)

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by device configuration business objects.

Business Object Options

Device configuration business objects can make use of the following business object options:

- **Synchronization Add BO:** This option identifies the business object to use when adding new device configurations as a result of a data synchronization request.

Example Device Configuration - D1-DeviceConfiguration

The table below lists the details of the D1-DeviceConfiguration device configuration business object.

Option	Description
Business Object	D1-DeviceConfiguration
Description	Device Configuration
Maintenance Object	D1-DVCCONFIG (Device Configuration)
Application Service	D1-DVCCONFIGBOAS (Device Configuration BO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none">• Synchronization Add BO: D1-SynchronizationAddDC (Device Configuration Synchronization Add)• Display UI Map: D1-DeviceConfigDisplay (Device Configuration- Display)• Portal Navigation Option: d1dvcfgTabMenu (Device Configuration)• Display Map Service Script: D1-D1-RtDvcCfgD (Device Configuration - Retrieve Details for Display)• Maintenance UI Map: D1-DeviceConfigMaint (Device Configuration- Maintenance)

Option	Description
Algorithms	<ul style="list-style-type: none">• Information: D1-DVCOINFO (Device Configuration Information)• Pre-Processing: D1-DELMC (Delete Associated Measuring Components)• Pre-Processing: D1-DEFTIMZON (Default Time Zone value based on Installation Option)• Validation: D1-INSTDVAL (Earlier Installed Device Configuration)• Validation: D1-VALTIMZON (Validates BO Time Zone value against Installation Option)
Lifecycle	<ul style="list-style-type: none">• Pending (Initial)• Active (Final)

Use the Business Object portal to view additional details concerning this business object.

Measuring Component Types In Detail

This section provides details concerning the measuring component type objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom measuring component type objects you create as part of your implementation. This section includes:

- A description of the D1-MCTYPE maintenance object
- Lists of the base package measuring component type business objects, including “lite” business objects
- Details concerning measuring component type-specific configuration options
- A sample measuring component type business object (D1-IntervalChannelTypePhysical)

Maintenance Object - D1-MCTYPE

Measuring component type business objects use the D1-MCTYPE maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-MCTYPE
Description	Measuring Component Type
Service Name	D1-MCTYPE (Measuring Component Type Maintenance)
Tables	<ul style="list-style-type: none">• D1_MEASR_COMP_TYPE (Measuring Component Type) - Primary• D1_MC_TYPE_VALUE_IDENTIFIER (Measuring Component Type Value Identifier) - Child• D1_MC_TYPE_VALUE_IDENTIFIER_L (MC Type Value Identifier Language) - Child• D1_MEASR_COMP_TYPE_CHAR (Measuring Component Type Characteristics) - Child• D1_MEASR_COMP_TYPE_L (Measuring Component Type Language) - Child• D1_MEASR_COMP_TYPE_VEE_GRP (Measuring Component Type VEE Group) - Child• D1_MEASR_COMP_TYPE_FBK_VEE_GRP (Measuring Component Type Fallback VEE Grp) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Measuring Component Type Business Objects

The Service and Measurement Data Foundation base package includes the following measuring component type business objects:

Business Object Name	Description
D1-ActStatsMasterAggType	Activity Statistics Master Aggregator Type
D1-ActStatsSubAggregatorType	Activity Statistics Sub-Aggregator Type
D1-ActivityStatsAggType	<p>Activity Statistics Aggregator Type</p> <p>Instances of this business object represent individual activity statistics aggregator measuring component types defined in the system.</p> <p>Activity Statistics Aggregators are used to track statistics related to device event and measurement uploads.</p>
D1-AutoReadRegisterType	<p>Auto-Read Register Type</p> <p>Instances of this business object represent individual auto-read register measuring component types defined in the system.</p> <p>Auto-Read Registers are automatically read on a regular basis but the system receives only one measurement at a time, meaning one measurement per Initial Measurement. These types of measuring components are categorized as scalar rather than interval, even though initial measurements are received on a regular basis.</p>
D1-GenericMCType	Generic MC Type (used as a parent measuring component type)
D1-IntervalChannelTypePhysical	<p>Interval Channel Type - Physical</p> <p>Instances of this business object represent individual physical interval channel measuring component types defined in the system.</p>
D1-IntervalChannelTypeScratchp	<p>Interval Channel Type - Scratchpad</p> <p>Instances of this business object represent individual scratchpad interval channel measuring component types defined in the system.</p>
D1-RegisterTypePhysical	<p>Register Type</p> <p>Instances of this business object represent individual physical register measuring component types defined in the system.</p>

The Service and Measurement Data Foundation base package includes the following “lite” measuring component type business objects:

Business Object Name	Description
D1-AggregatorTypeLite	D1-Aggregator Type Lite
D1-MCTypeEstimateParmsLiteBO	MC Type Periodic Estimation LITE
D1-MCTypeLite	MC Type Lite
D1-MCTypeMainLite	Measuring Component Type LITE
D1-MCTypeParentLITE	Measuring Component Type Parent LITE
D1-MeasuringCompTypeLite	Measuring Component Type LITE
D1-ReadMethodRegisterTypeLite	Register Type LITE BO

The Service and Measurement Data Foundation base package includes the following additional measuring component type business objects:

Business Object Name	Description
D1-MCTypeValueIdentifiers	Measuring Component Type Value Identifiers (used to retrieve the value identifiers of a measuring component type)
D1-MeasuringCompTypeBundlingBO	Bundling BO for Measuring Component Type
D1-MeasuringCompTypePhysicalBO	Physical BO for Measuring Component Type

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by measuring component type business objects.

System Events

Measuring component type business objects can make use of the following system events:

- **Calculate Interval Consumption:** This system event defines the algorithm used to calculate interval consumption for measuring components based on this measuring component type.
- **Calculate Scalar Consumption:** This system event defines the algorithm used to calculate scalar consumption for measuring components based of this type.

Other Options

Measuring component types define many attributes of the measuring components of that type. These options are specified when creating measuring component types based on a measuring component type business object, and include the following:

Value Identifiers

Value identifiers store the values of UOM, TOU, and SQI that identify the measured amounts for measuring components of this type. Value identifiers specify the quantities stored on the measurement records for measuring components of this type.

Value identifiers must also specify a Value Derivation Algorithm based on the D1-DERIVAQTY Algorithm Type (Derive a quantity using a formula). This algorithm is used to derive measurement elements by applying a formula to calculate a value.

VEE Groups (Valid and Fallback)

VEE Groups define the validation, editing, and estimation rules to be applied to initial measurement data for measuring components of this type.

- **Valid VEE Groups:** These define the VEE groups considered valid for measuring components of this type.
- **Fallback VEE Groups:** These define default VEE groups that can be used with all measuring components of this type. This alleviates the need to specify the same VEE groups on multiple measuring components of the same type. Each VEE group is designated a VEE group role that indicates when and how the VEE group is used (for initial load, manual override, or estimation).

Profile Factors

Profile factors are factors of type profile used when displaying initial measurement data on the Measuring Component portal of the 360 Degree View. Measuring component types reference the following types of profile factors:

- **Eligible Profile Factors (interval only):** These define the profile factors that are considered to be eligible for interval measuring components of this type. You can also specify one or more profile factors as a default.
- **Valid Profile Factors for Conversion from Scalar to Interval (scalar only):** These define the profile factors that are considered to be eligible for scalar measuring components of this type when converting scalar measurements to interval measurements. You can also specify one or more profile factors as a default.

Valid Scratchpad Measuring Component Types

These define the scratchpad measuring component types considered valid for measuring components of this type.

Display Properties

Measuring component types reference the following display properties:

- **Event Bar Profiles:** used when displaying measurement data for measuring components of this type. Event bar profiles are business objects defined as values for the “360 View Event Bar Profile” extendable lookup (D1-360EventBarProfile).
- **Final Values Overlay Profiles:** This display option is used when displaying final measurement data for measuring components of this type. Final values overlay profiles are business objects defined as values for the “Final Values Overlay Profile” extendable lookup (D1-FinalValuesOverlayProfile).

Example Measuring Component Type - D1-IntervalChannelTypePhysical

The table below lists the details of the D1-IntervalChannelTypePhysical measuring component type business object.

Option	Description
Business Object	D1-IntervalChannelTypePhysical
Description	Interval Channel Type - Physical
Maintenance Object	D1-MCTYPE (Measuring Component Type)
Application Service	D1-MCTYPE (Measuring Component Type MO)

Option	Description
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> • Display UI Map: D1-IntervalChannelTypeDisplay (Interval Channel Type - Display) • Portal Navigation Option: d1mctypeTabMenu (Measuring Component Type) • Maintenance UI Map: D1-IntervalChannelTypeMaint (Interval Channel Type - Maintenance)
Algorithms	<ul style="list-style-type: none"> • Calculate Interval Consumption: D1-IN-CNSUMP (Calculate Interval Consumption) • Validation: D1-MCTVIVAL (Validate UOM/TOU/SQI Combination) • Validation: D1-INTSCRVAL (Validate List of Scratchpad Measuring Component Types) • Validation: D1-DEFTOUVAL (Validate Interval Size of TOU Map) • Validation: D1-ELPRFTVAL (Validate Factors for Eligible Profile Factors) • Validation: D1-FNVALOVAL (Validate Final Values Overlay Profiles) • Validation: D1-EVTBARVAL (Validate Event Bar Profiles) • Validation: D1-HRDATAVAL (Hours of Data to Display Validation)

Use the Business Object portal to view additional details concerning this business object.

Meter Data Management Base Package Measuring Component Type Business Objects

Oracle Utilities Meter Data Management base package includes the following measuring component type business objects:

Business Object Name	Description
D2-AggregatorType	Aggregator Type Instances of this business object represent individual aggregator measuring component types defined in the system.
D2-ItemAggregatorType	Item Aggregator Type Instances of this business object represent individual item aggregator measuring component types defined in the system.
D2-MasterMsrmtAggregatorType	Master Measurement Aggregator Type Instances of this business object represent individual master measurement aggregator measuring component types defined in the system.
D2-SubMeasurementAggType	Sub Measurement Aggregator Type Instances of this business object represent individual sub measurement aggregator measuring component types defined in the system.

Oracle Utilities Meter Data Management base package includes the following “lite” measuring component type business objects:

Business Object Name	Description
D2-AggregatorTypeLite	Aggregator Type - Lite
D2-MCTypeLite1	Measuring Component Type Lite
D2-MCTypeScratchPads	Measuring Component Type Scratchpads LITE
D2-MCTypeValueIdentifiersLife	Measuring Component Type Value Identifiers LITE
D2-MQAggregatorTypeLite	Measurement Quantity Aggregator Type Lite
D2-ReadMethodRegisterTypeLite	Register Type Lite BO
D2-ServiceTypeMCTypeLite	Service Type MC Type List

Oracle Utilities Meter Data Management base package includes the following additional measuring component type business objects:

Business Object Name	Description
D2-MCTypeEligibleProfileFactor	Measuring Component Type Profile Factors (used to read the eligible profile factors of a measuring component type)

Business Object Name	Description
D2-ScalarMCTypeProfileFactors	Measuring Component Type Profile Factors (used to read the eligible profile factors used to convert scalar to interval.)

Measuring Components In Detail

This section provides details concerning the measuring components supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom measuring components you create as part of your implementation.

This section includes:

- A description of the D1-MEASRCOMP maintenance object
- Lists of the base package measuring component business objects, including “lite” business objects
- Details concerning measuring component-specific configuration options
- A sample measuring component business object (D1-IntervalChannel)
- Lists of base package measurement functions including the BPA Script, Service Script, and a brief description of each

Maintenance Object - D1-MEASRCOMP

Measuring component business objects use the D1-MEASRCOMP maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-MEASRCOMP
Description	Measuring Component
Service Name	D1-MEASRCOMP (Measuring Component Maintenance)
Tables	<ul style="list-style-type: none"> • D1_MEASR_COMP (Measuring Component) - Primary • D1_MEASR_COMP_CHAR (Measuring Component Characteristics) - Child • D1_MEASR_COMP_IDENTIFIER (Measuring Component Identifier) - Child • D1_MEASR_COMP_LOG (Measuring Component Log) - Child • D1_MEASR_COMP_LOG_PARM (Measuring Component Log Parameter) - Child • D1_MEASR_COMP_REL (Related Measuring Component) - Child • D1_MEASR_COMP_VEE_GROUP (Measuring Component VEE Group) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Measuring Component Business Objects

The Service and Measurement Data Foundation base package includes the following measuring component business objects:

Business Object Name	Description
D1-IntervalChannel	Interval Channel Instance of this business object represent individual interval channel measuring components defined in the system.
D1-IntervalScratchpad	Interval Scratchpad Instance of this business object represent individual interval scratchpad measuring components defined in the system.
D1-PayloadStatsAggEvent	Payload Statistics Aggregator Event Instance of this business object represent individual event payload statistics aggregator measuring components defined in the system.
D1-PayloadStatsAggIMD	Payload Statistics Aggregator - IMD Instance of this business object represent individual measurement payload statistics aggregator measuring components defined in the system.
D1-Register	Register Instance of this business object represent individual register measuring components defined in the system.
D1-RegsiterAutoRead	Register - Auto-Read Instance of this business object represent individual auto-read register measuring components defined in the system.
D1-StatsAggregator	Statistics Aggregator (used as a parent to payload statistics aggregators).

The Service and Measurement Data Foundation base package includes the following “lite” measuring component business objects:

Business Object Name	Description
D1-AggregatorLite	Aggregator LITE
D1-MCLatestMeasrDttm	Measuring Component Lite BO for Latest Measurement Date/Time
D1-MCLite	Measuring Component LITE
D1-MCScratchpadLite	MC Lite Scratchpad
D1-MCStandAlone	Measuring Component Standalone Lite
D1-MeasuringCompParentLITE	Measuring Component Parent LITE

The Service and Measurement Data Foundation base package includes the following additional measuring component business objects:

Business Object Name	Description
D1-SynchronizationAddMC	Measuring Component Synchronization Add (used when adding a new measuring component as a result of a data synchronization request)

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by measuring component business objects.

Business Object Options

Measuring component business objects can make use of the following business object options:

- **Estimation Initial Measurement Data BO:** This option identifies the initial measurement data business object to use when creating new “estimation” type initial measurement data for the measuring component.

For example, for the D1-IntervalChannel measuring component business object, this option is set to D1-EstimationIMDInterval, meaning that new “estimation” type initial measurement data for measuring components based on the D1-IntervalChannel business object would use the D1-EstimationIMDInterval business object.

- **Interval Initial Measurement Function:** This option defines the BPA Script used to apply a function to an interval initial measurement curve on the Initial Measurement Lens zone of the Initial Measurement portal. This option can be defined multiple times for the same measuring component.
- **Manual Override IMD BO:** This option identifies the initial measurement data business object to use when creating new “manual” type initial measurement data for the measuring component.

For example, for the D1-IntervalChannel measuring component business object, this option is set to D1-ManualIMDInterval, meaning that new “manual” type initial measurement data for measuring components based on the D1-IntervalChannel business object would use the D1-ManualIMDInterval business object.

- **Measuring Component Consumption Function:** This option defines the BPA Script used to apply a function to the measuring component's consumption on the zones of the Measuring Component portal in the 360 Degree View. This option can be defined multiple times for the same measuring component.
- **Synchronization Add BO:** This option identifies the business object to use when adding new measuring components as a result of a data synchronization request.

System Events

Measuring component business objects can make use of the following system events:

- **Find Constituent Measuring Components:** This system event defines the algorithm to use to identify constituent measuring components related to an aggregator measuring component. (An aggregator's constituent measuring components are the individual measuring components whose measurement data is aggregated when creating aggregation measurement data).
- **Periodic Estimation:** This system event defines the algorithm to use when performing periodic estimation for the measuring component.

Example Measuring Component - D1-IntervalChannel

The table below lists the details of the D1-IntervalChannel measuring component business object.

Option	Description
Business Object	D1-IntervalChannel
Description	Interval Channel
Maintenance Object	D1-MEASRCOMP (Measuring Component)
Application Service	D1-MEASURINGCOMP (Measuring Component MO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> • Estimation Initial Measurement Data BO: D1-EstimationIMDInterval (Estimation IMD (Interval)) • Manual Override IMD BO: D1-ManualIMDInterval (Manual IMD (Interval)) • Synchronization Add BO: D1-SynchronizationAddMC (Measuring Component Synchronization Add) • Measuring Component Consumption Functions: <ul style="list-style-type: none"> • D2-CScIntFnc (Create Scratchpad) • D2-IETExc (Export to Excel) • D2-NewIntFnc (New IMD via XML) • D2-OvrIntFnc (Create/Override) • D2-RedValFnc (Rederive Values) • D2-SAsIntFnc (Save As) • Interval Initial Measurement Functions: <ul style="list-style-type: none"> • D2-AdjMathF (Adjust Intervals Using Math) • D2-AdjScalF (Adjust Intervals to Scalar Quantity) • D2-IIMDSAsF (Save As) • D2-InsIntF (Insert Intervals) • D2-RemIntF (Remove Intervals) • D2-SetCondsF (Set Conditions) • D2-ShiftIntF (Shift Intervals) • D2-SmoSpikeF (Smooth Spikes) • Portal Navigation Option: d1mcTabMenu (Measuring Component) • Display Map Service Script: D1-RtMcIcChs (Interval Channel - Retrieve Details for Display)

Option	Description
Algorithms	<ul style="list-style-type: none"> Periodic Estimation: D1-CIITBIH (Create Interval IMD and To Do Based Upon Install History) Information: D1-SMTMCINFO (Measuring Component for Smart Devices - Information) Pre-Processing: D1-DIMDCPRE (Delete Initial Measurement Data of Measuring Component) Validation: D1-RELMCCVAL (Validate Consumption Reference Measuring Component) Validation: D1-INTMCVAL (Check if Measuring Component's Type is Interval)

Use the Business Object portal to view additional details concerning this business object.

Meter Data Management Base Package Measuring Component Business Objects

The Oracle Utilities Meter Data Management base package includes the following “lite” measuring component business objects:

Business Object Name	Description
D2-AggregatorLite	Aggregator Lite
D2-MCLite1	Measuring Component Attributes LITE
D2-RelatedMC	Related Measuring Components LITE (used to read the related measuring components of a measuring component)

The Oracle Utilities Meter Data Management base package includes the following additional measuring component business objects:

Business Object Name	Description
D2-Aggregator	Service Type and Postal Aggregator Instances of this business object represent individual service type and postal aggregator measuring components defined in the system.
D2-ConsumptionReferenceMC	Consumption Reference Measuring Component (used to read the consumption reference measuring component associated with interval measuring components)
D2-MeasuredQuantityAggregator	Measurement Measured Quantity Aggregator Instances of this business object represent individual measured quantity aggregator measuring components defined in the system.
D2-MsrmtQualityCountAggregator	Measurement Quality Count Aggregator Instances of this business object represent individual quality count aggregator measuring components defined in the system.

Business Object Name	Description
D2-MsrmtTimelinessCountAggr	Measurement Timeliness Count Aggregator Instances of this business object represent individual timeliness count aggregator measuring components defined in the system.
D2-MsrmtTimelinessQuantityAggr	Measurement Timeliness Quantity Aggregator Instances of this business object represent individual timeliness quantity aggregator measuring components defined in the system.

Base Package Interval Initial Measurement Functions

The following table lists the back package interval initial measurement functions. Each of these functions is implemented as a BPA Script and corresponding Service Script.

Function	BPA Script	Service Script	Description
Adjust Intervals to Scalar Quantity	D2-AdjScalF	D2-AdjScalS	Adjusts all interval values within a user-defined time period in an interval measurement such that the total of all interval values equals a user-defined scalar quantity.
Adjust Intervals Using Math	D2-AdjMathF	D2-AdjMathS	Adjusts all interval values within a user-defined time period using math operations (add, subtract, multiply, or divide).
Insert Intervals	D2-InsIntF	D2-InsIntS	Inserts intervals into initial measurement data.
Remove Intervals	D2-RemIntF	D2-RemIntBck	Removes intervals from initial measurement data
Save As	D2-IMDSAsF	D2-IMDSAs	Creates new initial measurement data for a measuring component other than that being displayed by copying all or some of the final measurements shown in the zone.
Set Conditions	D2-SetCondsF	D2-SetCondsS	Sets condition codes within initial measurement data. Condition codes indicate the circumstances (estimated, missing, etc.) of individual measurements. Condition codes are assigned to both scalar and interval measurement data both for initial measurement data and final measurements.
Shift Intervals	D2-ShfIntF	D2-ShfIntBck	Shifts intervals of initial measurement data to a new start date and time.
Smooth Spikes	D2-SmSpkF	D2-SmoSpikeS	Reduces "spike" intervals (intervals with values that are more than a user-defined percentage higher than other intervals within the initial measurement data).

Use the Script portal to view more details about these functions. The scripts listed above use a set of base package measurement services. See **Appendix A: Measurement Services** for a list of available base package measurement services.

Base Package Measuring Component Consumption Functions

The following table lists the back package interval initial measurement functions. Each of these functions is implemented as a BPA Script and corresponding Service Script.

Function	BPA Script	Service Script	Description
Create Scratchpad	D2-CScIntFnc	D2-CScIntBck	Creates a scratchpad measuring component along with new initial measurement data by copying all or some of the final measurements from the measuring component displayed in the zone.
Convert Scalar to Interval Function	D2-CnvSTIFnc	D2-CnvSTIBck	
Create/Override Interval Create/Override Scalar	D2-OvrIntFnc D2-OvrScaFnc	D2-OvrIntBck D2-OvrScaBck	Creates new initial measurement data for a selected measuring component for all or part of a selected time period. This function can either copy existing final measurements or create new measurement data for the initial measurement it creates.
Export to Excel	D2-IETxc D2-ScaETxc		Exports measurement data to a comma-separated values spreadsheet
New IMD via XML New Reading	D2-NewIntFnc D2-NewScaFnc	D2-NewIntBck D2-NewScaBck	Creates a new initial measurement reading (interval or scalar) for the measuring component.
Rederive Values	D2-RedValFnc	D2-RederVal	Rederives the measurement's values displayed in the zone.
Rederive Scalar Values:	D2-SRedVaFnc	D2-SRederVal	Rederives the measurement's values displayed in the zone.
Save As	D2-SAsIntFnc	D2-SAsIntBck	Creates new initial measurement data for a measuring component other than that being displayed by copying all or some of the final measurements shown in the zone.

Use the Script portal to view more details about these functions. The scripts listed above use a set of base package measurement services. See **Appendix A: Measurement Services** for a list of available base package measurement services.

Configuring Devices, Measuring Components, and Device Configurations

This section provides high-level overviews of the steps involved in configuring custom devices, device configurations, measuring component types, and measuring components. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

Note: The procedures below focus on specific configuration tasks and options related to each of the objects described in this chapter, and do not address all the steps involved in creating business objects and other configuration objects. For more information about these subjects, refer to the Oracle Utilities Application Framework documentation.

Configuring Custom Devices

Configuring custom devices involves the following steps:

1. Design the device business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom device-related configuration objects required for your business objects, including:

Business Object Options: Create business objects for the following business object options:

- Install Event BO
 - Synchronization Add BO
 - Valid Command Request BO (if using Oracle Utilities Smart Grid Gateway)
3. Create your device business objects, referencing the configuration objects created above as appropriate.
 4. Set up admin records that define the device types you will use in your implementation.

Configuring Custom Device Configurations

Configuring custom device configurations involves the following steps:

1. Design the device configuration business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom device configuration-related configuration objects required for your business objects, including:

Business Object Options: Create business objects for the following business object options:

- Synchronization Add BO
3. Create your device configuration business objects, referencing the configuration objects created above as appropriate.
 4. Set up admin records that define the device configuration types you will use in your implementation.

Configuring Custom Measuring Component Types

Configuring custom measuring component types involves the following steps:

1. Design the measuring component type business objects you will need to create for your implementation, including the data and processing required for each.

2. Create the custom measuring component type-related configuration objects required for your business objects, including:
System Events: Create algorithms for the following system events:
 - Calculate Interval Consumption (for interval and scalar measuring component types)
 - Calculate Scalar Consumption (for scalar measuring component types)**Options:** Create data as appropriate for the following options used when creating measuring component types:
 - Value Identifiers: Value Derivation Algorithms (based on the D1-DERIVAQTY Algorithm Type)
 - VEE Groups
 - Profile Factors
 - Scratchpad Measuring Component Types
 - Display Properties:
 - Event Bar Profiles: Values for the D1-360EventBarProfile extendable lookup
 - Final Values Overlay Profiles: Values for the D1-FinalValuesOverlayProfile extendable lookup.
3. Create your measuring component type business objects, referencing the configuration objects created above as appropriate.
4. Set up admin records that define the measuring component types you will use in your implementation.

Configuring Custom Measuring Components

Configuring custom measuring components involves the following steps:

1. Design the measuring component business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom measuring component-related configuration objects required for your business objects, including:
Business Object Options: Create business objects and algorithms for the following business object options:
 - Estimation Initial Measurement Data BO
 - Interval Initial Measurement Function
 - Manual Override IMD BO
 - Measuring Component Consumption Function
 - Synchronization Add BO
3. Create your measuring component business objects, referencing the configuration objects created above as appropriate.

Chapter 5

Service Points and Device Installation

This chapter provides descriptions of entities related to installation of meters, including service points, contacts, install events, activities, and other entities. This chapter includes:

- **Understanding Service Points and Device Installation**
- **Service Points In Detail**
- **Facilities In Detail**
- **Network Locations In Detail**
- **Contacts in Detail**
- **Install Events in Detail**
- **Service Providers in Detail**
- **Processing Methods in Detail**
- **Configuring Service Point and Device Installation Objects**

Understanding Service Points and Device Installation

This section provides an overview of entities related to the installation of devices (service points, facilities and network locations, contacts, install events, service providers, and others) how they are used in the Service and Measurement Data Foundation and related products, including Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway.

Service Points

Service points are physical locations at which a company supplies service. Devices are installed at service points. The relationship between individual service points and devices can change over time. For example, at any point in time:

- A service point may have a single device installed (or no device may be installed)
- A device may be installed at a single service point (or it may not be installed at a service points)

Over time:

- Different devices may be installed at a service point
- A device may be installed at different service points

Service points created from the base package service point business object (D1-ServicePoint) can be associated with facilities within the distribution network via the "Distribution Network Facility" field.

Service Point Categories

A service point's "Service Point Category" (defined for the service point's type) determines the type of devices that can be installed at that service point. The base package supports three service point categories:

- Meter: Indicates that a single meter can be installed at the service point
- Item: Indicates that a single "badged" item can be installed at the service point
- Multi-Item: Indicates that one or more "unbadged" items can be installed at the service point

An Aside: No Premise Object Exists

Oracle Utilities Meter Data Management (and other related meter data products) is not considered the system of record for premises or service points. The customer information system (or some other system) is considered the system of record for this type of information. In order to minimize the amount of data that needs to be synchronized, premise-oriented attributes used by meter data products are held on service points. This is an important distinction to keep in mind when creating custom service points for your implementation.

Facilities and Network Locations

Facilities represent the network node level closest to the service point. In terms of electric networks, facilities represent transformers.

Network locations define the location of a facility within a larger network.

Network locations specify the network nodes that provide service to a facility. In the case of electric networks, network locations define the feeder and substation associated with a transformer (facility). Because these network nodes can change over time, a facility can have many network locations (this is an effective-dated relationship). For example, for electric service points, whenever the substation or feeder that provides power to a given transformer is changed, a new record that references the latest substation and feeder should be created.

Note that feeders and substations are not directly represented in the system.

A key use of facility and network location information is in business intelligence applications (such as Oracle Utilities Analytics) to allow for analysis of consumption and device events based on network nodes. For example, using this data in a business intelligence application would allow viewing peak-time consumption for electric vehicle service points associated with a specific substation, feeder, and transformer.

When performing data synchronization to create extracts for business intelligence applications (such as Oracle Utilities Analytics), the following extendable lookups are used to map codes for facilities and nodes (substations, feeders and transformers) to descriptions:

- Facility Type (D1-FacilityTypeLookup)
- Network Node Type (D1-NetworkNodeTypeLookup)

Contacts

Contacts are individuals or business entities with which a company has contact. Service points can have associated contacts which define the individual or business entity that uses the service (electricity, gas, water, etc.) delivered at the service point. Note that while contacts are optional for service points, usage subscriptions must reference contacts.

Note: The base-package name search on the 360° Search looks for usage subscription-related contacts. Use the Service Point Query portal to find a service point using a service point-related contact.

Installation Events

Whenever a device is installed at a service point, an installation event is created. Installation events capture the history of the devices that have been installed at a service point. This allows consumption for a service point to be calculated over time. In technical terms, installation events (or install events) link a specific device configuration to a service point.

There are specific install event business objects for different types of devices, including smart meters, items, manual meters, and communication components.

When creating install events for items, only “badged” items (those that are uniquely identified in some way) used item install events. “Unbadged” items do not use install events. Instead, the number of each type of item is specified in the "Multi-Item Detail" section of the service point.

While a device is installed at a service point, it may be turned off (and back on again). The installation event that records the original installation date and time also records the dates and times when the device has been turned on and off. When a device is removed, the original installation event is updated with the removal date and time.

Service Providers

Service provider are external entities that serve various roles relative to the application. These can include head-end systems, billing systems to which the application sends bill determinant data, market participants in a deregulated environment, outage management systems that receive meter event data from the application, or other parties that require or provide information to the system.

Service providers can have one or more associated processing methods that define the format or means by which a service provider receives or sends data from or to the application, such as bill determinants, interval data, or meter events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and usage transactions. Processing methods can also be used to define the information an external system wishes to subscribe to receive from our application.

Service Providers as Head-End Systems

Head-end systems are systems that collect measurement data and meter events for eventual submission to the application. Many devices can communicate to the application through a single head-end system, but a utility may have numerous head-end systems through which they communicate with devices.

As noted above, head-end systems are defined as service providers. Head-end systems utilize processing methods that specify the type of initial measurement data and device events to create for devices (and their related measuring components) based on measuring component type.

Head-end system service providers also utilize processing methods that specify how smart meter commands are processed.

Service Providers as External Systems

External systems are applications and systems that are external to the Oracle Utilities meter data products, and can include customer information systems such as Oracle Utilities Customer Care and Billing, outage management systems such as Oracle Utilities Network Management System, or other types of applications.

External system service providers utilize processing methods to specify how the system sends/creates data used by the two applications. For example, when Oracle Utilities Meter Data Management is integrated with Oracle Utilities Customer Care and Billing, an external system representing Oracle Utilities Customer Care and Billing would specify how usage requests are received and processed by Oracle Utilities Meter Data Management,

Service Providers in Deregulated Markets

Some utilities operate in deregulated markets. In implementations in deregulated markets, the system can send information to and receive information from a variety of market entities. These entities are defined as service providers.

For example, a service point's distribution company and/or energy supply company may subscribe to its consumption, or a service point's meter service provider may send requests to ping the meter that's installed at the service point to verify connectivity between the meter and its head-end system.

Different Relationship Types In Different Markets

Each market can define different relationship types between its service providers. A single instance of Oracle Utilities Meter Data Management or Oracle Utilities Smart Grid Gateway may have service points in different markets where each market has different relationship types and service providers. For example:

- In a regulated market the distribution company is the de facto energy supplier and meter service provider.
- Another market might have two relationship types and a single service provider for each relationship:
 1. There is a single energy supply company for the entire market
 2. There is a single meter service provider for the entire market
- Yet another market might have two relationship types (energy supply and meter service). In this market, there might be multiple service providers for each relationship type. Each service point can choose any of the relationship type's service providers. If a service point does not declare a specific service provider for a given relationship type, the relationship type's "fallback" service provider is assumed.

Measurement Cycles

Measurement cycles define the schedule for manual meter reading of devices at service points. More specifically:

- A **measurement cycle** defines WHEN the service point is visited
- A **route** within a cycle defines a group of service points in a cycle that are visited by a meter reader
- A **sequence** within a route defines the physical position of the service point within a route
- A **schedule** specifies the dates on which service points are visited.

Manually read service point reference a measurement cycle, route and sequence within the route. A batch process creates SP/Measurement Cycle Schedule Routes for each service point, which link the dates on a measurement cycle schedule to a measurement cycle route defined for the service point. These define the specific date on which a meter reader will visit service points associated with a specific measurement cycle route, and the sequence in which the service points on that route are visited.

The **Route Management** portal allows users to manually manage the sequence of service points within a route, including renumbering the sequence of a user-selected set of service points and transferring one or more service to a different measurement cycle route.

Measurement Cycles can also be used to periodically push bill determinants to subscribing systems. See **Measurement Cycle And Creating Bill Determinants** below for more information.

Measurement Cycle Batch Processing

Measurement cycle processing is managed by the following three batch processes:

- Create Pending Measurement Cycle Schedule Routes (D1-CMCS)
This batch process creates Schedule Routes for Measurement Cycle Schedules whose schedule selection date is on or before the batch business date. This process is used if routes have the same schedule each month, quarter, etc. This process simply copies the routes from the Measurement Cycle to the Measurement Cycle Schedule on/after the scheduled selection date.
- Create Pending SP / Measurement Cycle Schedule Route Records (D1-CSPSR)
This batch process creates a “SP/Measurement Cycle Schedule Route” transaction for every service point in the Measurement Cycle Schedule Route that is ready for processing.
- Process Pending SP / Measurement Cycle Schedule Route Records (D1-PSPSR)
This batch process transitions the Pending “SP/Measurement Cycle Schedule Route” transactions to their Complete state. Custom algorithms can be configured to do any additional necessary work, such as creating a “Meter Read Download” activity. This custom algorithm would be configured as an Enter algorithm on the “Complete” state of the SP/Measurement Cycle Schedule Route business object.

Measurement Cycle And Creating Bill Determinants

The system can be configured to periodically push bill determinants to subscribing systems. In this case, measurement cycles can be configured to define when to create usage transactions for usage subscriptions associated to service points in the cycle. In this case, even service points whose meters are read automatically may reference measurement cycles.

Creating bill determinants (by creating a usage transaction) is performed by an algorithm on the “Complete” state of the SP/Measurement Cycle Schedule Route business object (similar to creating activities as described above).

When the Pending SP/Measurement Cycle Schedule Route records are processed by the “Process Pending SP / Measurement Cycle Schedule Route Records” process (D1-PSPSR), rather than create a handheld download activity, the algorithm can create a usage transaction (usage transactions are transactions that cause bill determinants to be calculated for the service point’s usage subscription(s)).

If the implementation needs to both manually read the meter and push bill determinants, both algorithms would be plugged in on the SP/Measurement Cycle Schedule Route business object.

Service Points In Detail

This section provides details concerning the service point objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom service point objects you create as part of your implementation. This section includes:

- A description of the D1-SP maintenance object
- Lists of the base package service point business objects, including “lite” business objects
- Details concerning service point-specific configuration options
- A sample service point business object (D1-ServicePoint)

Maintenance Object - D1-SP

Service point business objects use the D1-SP maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-SP
Description	Service Point
Service Name	D1-SP (Service Point Maintenance)
Tables	<ul style="list-style-type: none"> • D1_SP (Service Point) - Primary • D1_SP_REL (Service Point Relationship) - Child • D1_SP_CHAR (Service Point Characteristics) - Child • D1_SP_CONTACT (Service Point Contact) - Child • D1_SP_FACILITY (Service Point Facility) - Child • D1_SP_IDENTIFIER (Service Point Identifier) - Child • D1_SP_LOG (Service Point Log) - Child • D1_SP_LOG_PARM (Service Point Log Parameter) - Child • D1_SP_MKT_PARTICIPANT (Service Point Market Participant) - Child • D1-SP_REL (Service Point Relationship) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Service Point Business Objects

The Service and Measurement Data Foundation base package includes the following service point business objects:

Business Object Name	Description
D1-ServicePoint	Service Point Instances of this business object represent individual service points defined in the system.

The Service and Measurement Data Foundation base package includes the following “lite” service point business objects:

Business Object Name	Description
D1-SPForCEUpd	SP For Completion Event Update
D1-SPLite	Service Point Lite
D1-ServicePointMultiItemLITE	Service Point Multi-Item LITE
D1-ServicePointODMBORead	Service Point ODM BO to Read
D1-ServicePointParentLITE	Service Point Parent LITE

The Service and Measurement Data Foundation base package includes the following additional service point business objects:

Business Object Name	Description
D1-SynchronizationAddSP	Service Point Synchronization Add (used when adding a new service point as a result of a data synchronization request)

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by service point business objects.

Business Object Options

Service point business objects can make use of the following business object options:

- **Synchronization Add BO:** This option identifies the business object to use when adding new service points as a result of a data synchronization request.

System Events

Service point business objects can make use of the following system events:

- **Service Point Snapshot:** This system event defines the algorithm used to create a snapshot of the service point for use with Oracle Utilities Business Intelligence. The Algorithm Entity for available algorithms is “SP (BO) - Snapshot.”
- **Usage Snapshot:** This system event defines the algorithm used to create a usage snapshot of the service point for use with Oracle Utilities Business Intelligence. The Algorithm Entity for available algorithms is “SP (BO) - Usage Snapshot.”
- **Unreported Usage Analysis Snapshot:** This system event defines the algorithm used to create a snapshot of the service point’s consumption since the last usage transaction for use with Oracle Utilities Business Intelligence. The Algorithm Entity for available algorithms is “SP (BO) - Unreported Usage Analysis Snapshot.”
- **SP VEE Exception Snapshot:** This system event defines the algorithm used to create a snapshot of VEE exceptions for the service point for use with Oracle Utilities Business Intelligence. The Algorithm Entity for available algorithms is “SP (BO) - VEE Exception Snapshot.”

Example Service Point - D1-ServicePoint

The table below lists the details of the D1-ServicePoint service point business object.

Option	Description
Business Object	D1-ServicePoint
Description	Service Point
Maintenance Object	D1-SP (Service Point)
Application Service	D1-SPBOAS (Service Point BO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> Synchronization Add BO: D1-SynchronizationAddSP (Service Point Synchronization Add) Portal Navigation Option: d1spTabMenu (Service Point) Pre-Processing Service Script: D1-SPMtnPre (Service Point Maintenance Pre-Processing) Display Map Service Script: D1-SPRetDts (Service Point - Retrieve Details for Display)
Algorithms	<ul style="list-style-type: none"> Service Point Snapshot: D1-SPSNAP-SE (SP Snapshot - SP Snapshot System Event) *Usage Snapshot: D2-SP-CA (Aggregate SP Usage Snapshot) *Unreported Usage Analysis Snapshot: D2-SP-UT-AGE (Analyze Unreported Usage) *SP VEE Exception Snapshot: D2-SPVEEEXC (SP VEE Exception Aggregator) Information: D1-SPINFO (Service Point Information) Pre-Processing: D1-DEFTIMZON (Default Time Zone Value based on Installation Option) Validation: D1-SPADDRVAL (Validate Service Point's Address Information) Validation: D1-MSCYCVAL (Validate Service Point's Measurement Cycle Information) Validation: D1-VALTIMZON (Validate BO Time Zone Against Installation Option) *Validation: D2-SPUS-VAL (Validate Primary Usage Subscription) Validation: D1-SVCTYPVAL (Facility Service Type Validation) Validation: D1-SPTYPCVAL (Service Point Type Category Validation) Validation: D1-MIPERDVAL (Multi-Item Period Validation) <p>* Available with Oracle Utilities Meter Data Management.</p>
Lifecycle	<ul style="list-style-type: none"> Active (Initial) Inactive (Final)

Use the Business Object portal to view additional details concerning this business object.

Meter Data Management Based Package Service Point Business Objects

The Oracle Utilities Meter Data Management base package includes the following “lite” service point business objects:

Business Object Name	Description
D2-SPEExternalID	Service Point's External ID LITE
D2-SPLite1	Service Point Time Zone
D2-SPPPrimaryUSLite	Service Point Primary US Lite

The Oracle Utilities Meter Data Management base package includes the following additional service point business objects:

Business Object Name	Description
D2-SPMainContact	Service Point's Main Contact (used to retrieve the main contact of a service point)

Facilities In Detail

This section provides details concerning the facilities objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom facility objects you create as part of your implementation. This section includes:

- A description of the D1-FACILITY maintenance object
- Lists of the base package facility business objects, including “lite” business objects
- Details concerning facility-specific configuration options
- A sample facility business object (D1-Transformer)

Maintenance Object - D1-FACILITY

Facility business objects use the D1-FACILITY maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-FACILITY
Description	Facility
Service Name	D1-FACILITY (Facility Maintenance)
Tables	<ul style="list-style-type: none"> • D1-FACILITY (Facility) - Primary • D1-FACILITY_CHAR (Facility Characteristic) - Child • D1-FACILITY_IDENTIFIER (Facility Identifier) - Child • D1-FACILITY_L (Facility Log) - Child • D1-FACILITY_LOG (Facility Log) - Child • D1-FACILITY_LOG_PARM (Facility Log Parameter) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Facility Business Objects

The Service and Measurement Data Foundation base package includes the following facility business objects:

Business Object Name	Description
D1-Transformer	Transformer Instances of this business object represent individual transformers defined in the system.

Facility-Related Extendable Lookups

The Service and Measurement Data Foundation base package includes the following facility-related extendable lookup business objects:

Business Object Name	Description
D1-FacilityTypeLookUp	Facility Type (used to map facility codes to descriptions when performing data extract processing for use with business intelligence applications such as Oracle Utilities Analytics)

Example Facility- D1-Transformer

The table below lists the details of the D1-Transformer facility business object.

Option	Description
Business Object	D1-Transformer
Description	Transformer
Maintenance Object	D1-FACILITY (Facility)
Application Service	D1-FCLTYBOAS (Facility BO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none">Portal Navigation Option: d1fcltyTabMenu (Facility Maintenance)
Algorithms	<ul style="list-style-type: none">Information: D1-TRNSFINFO (Transformer Information)Validation: D1-FCLTSTVAL (Facility Service Type Validation)
Lifecycle	<ul style="list-style-type: none">Active (Initial)Inactive (Final)

Use the Business Object portal to view additional details concerning this business object.

Network Locations In Detail

This section provides details concerning the network locations objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom network location objects you create as part of your implementation. This section includes:

- A description of the D1-NWLOC maintenance object
- Lists of the base package network location business objects, including “lite” business objects
- Details concerning network location-specific configuration options
- A sample network location business object (D1-ElectricityNetworkLocation)

Maintenance Object - D1-NWLOC

Network location business objects use the D1-NWLOC maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-NWLOC
Description	Network Location
Service Name	D1-NWLOC (Network Location Maintenance)
Tables	<ul style="list-style-type: none"> • D1-NWC (Network Location) - Primary • D1-NW_CHAR (Network Location Characteristic) - Child • D1-NW_IDENTIFIER (Network Location Identifier) - Child • D1-NWLLOG (Network Location Log) - Child • D1-NW_LOG_PARM (Network Location Log Parameter) - Child • D1-NW_NODE (Network Node) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Network Location Business Objects

The Service and Measurement Data Foundation base package includes the following network location business objects:

Business Object Name	Description
D1-ElectricityNetworkLocation	<p>Network Location</p> <p>Instances of this business object represent individual network locations defined in the system.</p>

Network Location-Related Extendable Lookups

The Service and Measurement Data Foundation base package includes the following network location-related extendable lookup business objects:

Business Object Name	Description
D1-NetworkNodeTypeLookup	Network Node Type (used to map network node codes to descriptions when performing data extract processing for use with business intelligence applications such as Oracle Utilities Analytics)

Example Network Location - D1-ElectricityNetworkLocation

The table below lists the details of the D1-Transformer facility business object.

Option	Description
Business Object	D1-ElectricityNetworkLocation
Description	Electricity Network Location
Maintenance Object	D1-NWLOC (Facility)
Application Service	D1-NWLOC (Network Location MO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none">Portal Navigation Option: d1nwlocTabMenu (Network Location)
Algorithms	<ul style="list-style-type: none">Information: D1-NWLOCINFO (Electricity Network Location Information)Validation: D1-FCLTYVAL (Facility Status Validation)Validation: D1-NWLDTVL (Network Location Date/Time Validation)

Use the Business Object portal to view additional details concerning this business object.

Contacts in Detail

This section provides details concerning the contact objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom contact objects you create as part of your implementation. This section includes:

- A description of the D1-CONTACT maintenance object
- Lists of the base package contact business objects
- A sample contact business object (D1-Business)

Maintenance Object - D1-CONTACT

Contact business objects use the D1-CONTACT maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-CONTACT
Description	Contact
Service Name	D1-CONTACT (Contact Maintenance)
Tables	<ul style="list-style-type: none"> • D1_CONTACT (Contact) - Primary • D1_CONTACT_CHAR (Contact Characteristics) - Child • D1_CONTACT_EMAIL (Contact Email) - Child • D1_CONTACT_IDENTIFIER (Contact Identifier) - Child • D1_CONTACT_NAME (Contact Name) - Child • D1_CONTACT_PHONE (Contact Phone) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Contact Business Objects

The Service and Measurement Data Foundation base package includes the following contact business objects:

Business Object Name	Description
D1-Business	Business Instances of this business object represent individual business contacts defined in the system.
D1-Person	Person Instances of this business object represent individual person contacts defined in the system.

The Service and Measurement Data Foundation base package includes the following “lite” contact business objects:

Business Object Name	Description
D1-ContactLITE	Usage Transaction To Do Contact LITE
D1-ContactODMBORRead	Contact ODM BO to Read

The Service and Measurement Data Foundation base package includes the following additional contact business objects:

Business Object Name	Description
D1-SynchronizationAddContact	Contact Synchronization Add (used when adding a new contact as a result of a data synchronization request)

Example Contact - D1-Business

The table below lists the details of the D1-Business contact business object.

Option	Description
Business Object	D1-Business
Description	Business
Maintenance Object	D1-CONTACT (Contact)
Application Service	D1-CONTACT (Contact MO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> Synchronization Add BO: D1-SynchronizationAddContact (Contact Synchronization Add) Portal Navigation Option: d1contctTabMenu (Contact)
Algorithms	<ul style="list-style-type: none"> Information: D1-BUS-INFO (Business Contact - Information)

Use the Business Object portal to view additional details concerning this business object.

Install Events in Detail

This section provides details concerning the install event objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom install event objects you create as part of your implementation. This section includes:

- A description of the D1-INSTLEVT maintenance object
- Lists of the base package install event business objects, including “lite” business objects
- A sample install event business object (D1-SmartMeterInstallEvent)

Maintenance Object - D1-INSTLEVT

Install event business objects use the D1-INSTLEVT maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-INSTLEVT
Description	Install Event
Service Name	D1-INSTLEVT (Install Event Maintenance)
Tables	<ul style="list-style-type: none"> • D1_INSTLEVT (Install Event) - Primary • D1_INSTLEVT_CHAR (Install Event Characteristics) - Child • D1_INSTLEVT_LOG (Install Event Log) - Child • D1_INSTLEVT_LOG_PARM (Install Event Log Parameter) - Child • D1_ON_OFF_HIST (On/Off History) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Install Event Business Objects

The Service and Measurement Data Foundation base package includes the following install event business objects:

Business Object Name	Description
D1-CommComponentInstallEvent	Communication Component Install Event Instances of this business object represent individual communication component installation events defined in the system.
D1-ItemInstallEvent	Item Installation Event Instances of this business object represent individual item installation events defined in the system.

Business Object Name	Description
D1-ManualMeterInstallEvent	Manual Meter Installation Event Instances of this business object represent individual manual meter installation events defined in the system.
D1-SmartMeterInstallEvent	Smart Meter Installation Event Instances of this business object represent individual smart meter installation events defined in the system.

The Service and Measurement Data Foundation base package includes the following “lite” install event business objects:

Business Object Name	Description
D1-CommCompInstallEventLITE	Communication Component Install Event LITE
D1-InstallEventBORead	ODM Install Event BO to Read
D1-InstallEventLite	Install Event Lite
D1-InstallEventMainLite	Install Event Main LITE
D1-InstallEventParentLITE	Install Event Parent LITE
D1-SmartMeterInstallEventLite	Smart Meter Install Event LITE

The Service and Measurement Data Foundation base package includes the following additional install event business objects:

Business Object Name	Description
D1-SynchronizationAddIE	Install Event Synchronization Add (used when adding a new installation event as a result of a data synchronization request)

Example Install Event - D1-SmartMeterInstallEvent

The table below lists the details of the D1-SmartMeterInstallEvent install event business object.

Option	Description
Business Object	D1-SmartMeterInstallEvent
Description	Smart Meter Installation Event
Maintenance Object	D1-INSTLEVT (Install Event)
Application Service	D1-SMTMTRINSEVTBOAS (Smart Event Installation Event BO)
Instance Control	Allow New Instances

Option	Description
Options	<ul style="list-style-type: none"> • Synchronization Add BO: D1-SynchronizationAddIE (Install Event Synchronization Add) • Display UI Map: D1-SmartMeterInstallEventDisplay (Smart Meter Install Event - Display) • Portal Navigation Option: d1inevtmTabMenu (Install Event) • Display Map Service Script: D1-SmtIERtDt (Smart Meter Install Event - Retrieve Details for Display) • Maintenance UI Map: D1-SmartMeterInstallEventMaint (Smart Meter Install Event - Maintenance)
Algorithms	<ul style="list-style-type: none"> • Information: D1-INEVTINFO (Install Event Information) • Pre-Processing: D1-DFLTINSTC (Default the Install Event's Installation Constant) • Pre-Processing: D1-POPARMSTS (Default the Arming Status) • Validation: D1-DEVCFGVAL (Validate the Device Configuration) • Validation: D1-ONHISTVAL (Validate the On/Off History based on the Previous Install Event) • Validation: D1-CHKHISEVT (Validate the On/Off History based on the Install Event's Status) • Validation: D1-CKIFOVLEX (Validate Overlapping Install Events) • Validation: D1-VALIERMDT (Validate Removal Information) • Validation: D1-VALDVSPT (Meter Service Point Type Category Validation)
Lifecycle	<ul style="list-style-type: none"> • Pending (Initial) • Connected/Pre-Commissioned (Interim) • Pre-Connected/Commissioned (Interim) • Connected/Commissioned (Interim) • Connected/Decommissioned (Interim) • Disconnected/Commissioned (Interim) • Disconnected/Decommissioned (Interim) • Remove (Final)

Use the Business Object portal to view additional details concerning this business object.

Service Providers in Detail

This section provides details concerning the service provider objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom service provider objects you create as part of your implementation. This section includes:

- A description of the D1-SVCPROVDR maintenance object
- Lists of the base package service provider business objects, including “lite” business objects
- Details concerning service provider-specific configuration options
- A sample service provider business object (D1-HeadEndSystem)

Maintenance Object - D1-SVCPROVDR

Service provider business objects use the D1-SVCPROVDR maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-SVCPROVDR
Description	Service Provider
Service Name	D1-SVCPROVDR (Service Provider Maintenance)
Tables	<ul style="list-style-type: none">• D1_SPR (Service Provider) - Primary• D1_SPR_CHAR (Service Provider Characteristics) - Child• D1_SPR_L (Service Provider Language) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Service Provider Business Objects

The Service and Measurement Data Foundation base package includes the following service provider business objects:

Business Object Name	Description
D1-ExternalApplication	External Application Instances of this business object represent individual external applications defined in the system.
D1-HeadEndSystem	Head-End System Instances of this business object represent individual head-end systems defined in the system.
D1-MarketParticipant	Market Participant Instances of this business object represent individual market participants defined in the system.

Business Object Name	Description
D1-ServiceProvider	Service Provider (generic service provider business object used as a parent business object for all other service provider business objects)

The Service and Measurement Data Foundation base package includes the following “lite” service provider business objects:

Business Object Name	Description
D1-MarketParticipantLite	Market Participant Lite
D1-MarketParticipantParentLite	Market Participant Parent Lite

The Service and Measurement Data Foundation base package includes the following additional service provider business objects:

Business Object Name	Description
D1-ServiceProviderBundlingAdBO	Bundling Add BO for Service Provider
D1-ServiceProviderPhysicalBO	Physical BO for Service Provider

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by service provider business objects.

Business Object Options

Service provider business objects can make use of the following business object options:

- **Service Provider Type:** This option defines the type of service provider. Valid values are defined as values for the SPR_TYPE_FLG lookup field. The base package includes the following options:

Lookup Field Value	Description
D1EA	Edge Application
D1HE	Head-End System
D1MP	Market Participant

Example Service Provider - D1-HeadEndSystem

The table below lists the details of the D1-HeadEndSystem service provider business object.

Option	Description
Business Object	D1-HeadEndSystem
Description	Head-End System
Maintenance Object	D1-SVCPROVDR (Service Provider)
Parent Business Object	D1-ServiceProvider

Option	Description
Application Service	D1-SVCPROVIDER (Service Provider MO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none">• Service Provider Type: D1HE (Head-End)• Display UI Map: D1-HeadEndSystemDisplay (Head-End System - Display)• Portal Navigation Option: d1svcproTabMenu (Service Provider)• Maintenance UI Map: D1-HeadEndSystemMaint (Head-End System - Maintenance)

Use the Business Object portal to view additional details concerning this business object.

Processing Methods in Detail

This section provides details concerning the processing method objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom processing method objects you create as part of your implementation. This section includes:

- A description of the D1-PROCMETHOD maintenance object
- Lists of the base package processing method business objects, including “lite” business objects
- Details concerning processing method-specific configuration options
- A sample service processing method object (D1-HowToCreateMCInformation)

Maintenance Object - D1-PROCMETHOD

Processing method business objects use the D1-PROCMETHOD maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-PROCMETHOD
Description	Processing Method
Service Name	D1-PROCMETHOD (Processing Method Maintenance)
Tables	<ul style="list-style-type: none"> • D1_PROC_MTHD (Processing Method) - Primary • D1_PROC_MTHD_CHAR (Processing Method Characteristics) - Child • D1_PROC_MTHD_L (Processing Method Language) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Business Objects

The Service and Measurement Data Foundation base package includes the following processing method business objects:

Business Object Name	Description
D1-AbstractProcessingMethod	Generic Processing Method (generic processing method business object used as a parent business object for all other processing method business objects)
D1-HowToCreateActivityOBComm	How to Create Outbound Communication / Send OB Message (used to determine the specific type of outbound communication/ message to send)
D1-HowToCreateMCInformation	How To Create MC Related Information (used to define how measuring component-related information is created for the service provider, including initial measurement data)

Business Object Name	Description
D1-HowToProcDvceEvtsInformation	How to Process Device Event Related Information (used to send device events to a service provider)
D1-HowToProcSPRelatedInfo	How to Process Service Point Related Info (used to allow for differentiation of processing methods based on service point type, along with a default processing method)
D1-HowToProcessDeviceInfo	How to Process Device Related Information (used to define data translations for head-end systems, including how UOM codes are mapped for devices of the head-end system, how device events are mapped, how time zones are translated, etc.)
D1-HowToProcSPRelatedInfo	How to Process Service Point Related Info (used to define how service point-related information is sent to the service provider)
D1-HowToSendActInformation	How to Send Activity Related Information (used to define how activity-related information is sent to the service provider)
D1-HowToSendActivityRemarkInfo	How to Send Field Activity Remark Information (used to define how to send field activity remark information to the service provider)
D1-HowToSendActivityResponse	How to Send Activity Related Outbound Messages (used to define the outbound message type sent to a service provider in response to an activity)
D1-HowToSendFARelatedInfo	How to Send Field Activity Related Info (used to define how field activity related information is sent to the service provider)

The Service and Measurement Data Foundation base package includes the following additional processing method business objects:

Business Object Name	Description
D1-ProcessingMethodBundlingABO	Bundling Add BO for Processing Method
D1-ProcessingMethodPhysicalBO	Physical BO for Processing Method

Configuration Options

This section outlines specific types of BO Options and System Events used by processing method business objects.

Business Object Options

Processing method business objects can make use of the following business object options:

- **Applicable Processing Role:** This option defines the processing role for the processing method. Valid values are defined as values for the PROC_ROLE_FLG lookup field. The base package includes the following options:

Lookup Field Value	Description
D1AM	Obtain AMI Device Identifier
D1AR	Appointment Request
D1BR	Bulk Response
D1CA	Cancellation Activity
D1CC	Customer Contact
D1CD	Collection Details
D1DA	Activity Notification
D1DC	Device Commission
D1DD	Device Decommission
D1DM	Device Event Mapping
D1DR	Demand Reset
D1DS	Device Status Check
D1EP	Event Processing Default Configuration
D1FA	Field Activity
D1FC	Field Activity Completion
D1FR	Response - Fail
D1IM	Initial Measurement Creation
D1IN	On-Demand Read (Interval)
D1IS	Interim Status Update
D1LC	Load Check
D1ME	Meter Exchange Mapping
D1MS	Multi-Device Status Check
D1RA	Response - Appointment
D1RC	Remote Connect
D1RD	Remote Disconnect
D1RE	Response - Negative Acknowledgement
D1RM	Response - Missed Appointment
D1RN	Read Notification
D1RR	Response - Received
D1RT	Send Field Activity Remark
D1SC	On-Demand Read (Scalar)

Lookup Field Value	Description
D1SD	Send Device Event
D1SN	Suppression Notification
D1SQ	SQI Translation
D1SR	Response - Success
D1TU	TOU Translation
D1TZ	Time Zone Translation
D1UM	UOM Translation
D1UP	Update Activity
D1VC	Verify Commission

System Events

Processing method business objects can make use of the following system events:

- Determine Processing Method(s):** This system event defines the algorithm used to determine the processing methods (business object or batch code) to use, based on the related entity. The Algorithm Entity for available algorithms is “Proc Method (BO) - Determine Proc Method.” The base package includes the following algorithm types and algorithms for use with this system event:

Algorithm Type / Algorithm	Description
D1-BODIFFAT / D1-BODIFFAT	BO / Batch Differs By Activity Type (returns the BO code or batch code used to send activity-related information to a service provider for a processing role)
D1-BODIFFAT / D1-BOBDIFFAT	BO / Batch Differs By Activity Type (returns the BO code or batch code used to send activity-related information to a service provider for a processing role)
D1-BODIFFDT / D1-BODIFFDT	BO Differs by Device Type (returns the BO code used in the creation of device-oriented information for a service provider and processing role)
D1-BODIFFMCT / D1-BODIFFMCT	BO Differs By Measuring Component Type (returns the BO code used to create measuring component-related information for a service provider and processing role)
D1-BODIFFSPT / D1-BODIFFSPT	BO Differs By Service Point Type (returns the BO codes used to create service point-related information for a service provider and processing role)
D1-DIFDVETC / D1-DIFDVETC	Method Differs By Device Event Category (returns the BO or outbound message type or batch process used to send device events to a service provider)

Algorithm Type / Algorithm	Description
D1-OCDDT / D1-OCDDT	Outbound Communication Differs by Device Type (returns the BO used to send messages to a service provider)
D1-OMTDAT / D1-OMTDAT	Outbound Message Type Differs by Activity Type (returns the outbound message type used to create outbound messages for a service provider and processing role or message category/number if outbound message creation is not supported)
D1-OMTDFIT / D1-OMTDFIT	Outbound Message Type Differs by Field Task Type (returns the outbound message type for a service provider and processing role)
D1-PMDELV / D1-PMDELV	Processing Method Differs by Extendable Lookup Value (Opt-In) (determines the service providers for the processing method by examining the list of extendable lookup values for a given service provider)

Example Processing Method - D1-HowToCreateMCInformation

The table below lists the details of the D1-HowToCreateMCInformation business object.

Option	Description
Business Object	D1-HowToCreateMCInformation
Description	How to Create MC Related Information
Maintenance Object	D1-PROC METHD (Processing Method)
Parent Business Object	D1-AbstractProcessingMethod
Lifecycle Business Object	Generic Processing Method
Application Service	D1-PROC METHD (Processing Method MO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> Applicable Processing Role: D1IM Portal Navigation Option: d1svcproTabMenu (Service Provider) Maintenance UI Map: D1-HowToCreateMCInfoMaint (How To Crt MC Related Info - Maintenance)
Algorithms	<ul style="list-style-type: none"> Determine Processing Method(s): D1-BODIFFMCT (BO Differs by Measuring Component Type) Validation: D1-PROMTHVAL (Validate Processing Method)

Use the Business Object portal to view additional details concerning this business object.

Meter Data Management Base Package Processing Method Business Objects and Configuration Options

The Oracle Utilities Meter Data Management base package includes the following processing method business objects:

Business Object Name	Description
D2-HowToCreateUSInformation	How To Create US Related Information
D2-HowToSendUSInfoBatch	How To Send US Related Information Batch - DEPRECATED
D2-HowToSendUSInfoOnline	How To Send US Information Online - DEPRECATED
D2-HowToSendUSInformation	How To Send US Related Information

Business Object Options

Service provider business objects can make use of the following business object options:

- **Applicable Processing Role:** This option defines the processing role for the processing method. Valid values are defined as values for the PROC_ROLE_FLG lookup field. The base package includes the following options:

Lookup Field Value	Description
D2EB	Usage Trans Error Notification - Batch
D2EO	Usage Trans Error Notification - Online
D2UB	Usage Transaction Notification - Batch
D2UC	Usage Transaction Creation
D2UO	Usage Transaction Notification - Online
D2US	Usage Trans Subsequent Correction Notification
D2UT	Usage Transaction Notification

System Events

Service provider business objects can make use of the following system events:

- **Determine Processing Method(s):** This system event defines the algorithm used to determine the processing methods (business object or batch code) to use, based on the related entity. The Algorithm Entity for available algorithms is “Proc Method (BO) - Determine Proc Method.” The base package includes the following algorithm types and algorithms for use with this system event:

Algorithm Type / Algorithm	Description
D2-BOBDIFFUS / D2-BOBDIFFUS	BO / Batch Differs By Usage Subscription Type (returns the BO code or batch code used to send usage subscription-related information to a service provider for a processing role.)

Algorithm Type / Algorithm	Description
D2-BODIFFUST / D2-BODIFFUST	BO Differs By Usage Subscription Type (returns the BO code used to create usage subscription-oriented information for a service provider and processing role)

Configuring Service Point and Device Installation Objects

This section provides high-level overviews of the steps involved in configuring custom service points, contacts, install events, service providers, and activities. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

Note: The procedures below focus on specific configuration tasks and options related to each of the objects described in this chapter, and do not address all the steps involved in creating business objects, UI maps, algorithms, etc. For more information about these subjects, refer to the Oracle Utilities Application Framework documentation.

Configuring Custom Service Points

Configuring custom service points involves the following steps:

1. Design the service point business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom service point-related configuration objects required for your business objects, including:
Business Object Options: Create algorithms for the following business object options:
 - Process Measurement Cycle
3. Create your service point business objects, referencing the configuration objects created above as appropriate.
4. Set up admin records that define the service point types you will use in your implementation.

Configuring Custom Contacts

Configuring custom contacts involves the following steps:

1. Design the contact business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom contact-related configuration objects required for your business objects.
3. Set up admin records that define the contact types you will use in your implementation.

Configuring Custom Install Events

Configuring custom install events involves the following steps:

1. Design the install event business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom install event-related configuration objects required for your business objects.

Configuring Custom Service Providers

Configuring custom service providers involves the following steps:

1. Design the service provider business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom service provider-related configuration objects required for your business objects, including:

Processing Methods: Create processing method business objects for use with each service provider.

3. Create your service provider business objects, referencing the configuration objects created above as appropriate.

Note: Service provider business objects should reference D1-ServiceProvider as their Parent Business Object.

Configuring Custom Processing Methods

Configuring custom processing methods involves the following steps:

1. Design the processing method business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom processing method-related configuration objects required for your business objects, including:

System Events: Create algorithms for the following system events:

- Determine Processing Method(s)

3. Create your processing method provider business objects, referencing the configuration objects created above as appropriate.

Note: Processing method business objects should reference D1-AbstractProcessingMethod as their Parent Business Object.

Chapter 6

Measurement Data

This chapter provides descriptions of initial and final measurement data, including:

- **Understanding Initial Measurement Data and Final Measurements**
- **Understanding Estimated Initial Measurements**
- **Related Measuring Component Consumption Synchronization**
- **Initial Measurement Data In Detail**
- **Measurements In Detail**
- **Device Type Service Quantities in Detail**
- **Configuring Custom Measurement Data**

Understanding Initial Measurement Data and Final Measurements

This section provides an overview of initial measurements and final measurements and how they are used in Oracle Utilities Service and Measurement Data Foundation and Oracle Utilities Meter Data Management, including:

- **Initial Measurement Data**
- **Final Measurements**
- **Daylight Saving Time Support**

Initial Measurement Data

Measurements read from a measuring component are referred to as “initial measurement data” (or initial measurements) and are used to record how much of the quantity (defined by UOM, TOU, and SQI) measured by the measuring component was consumed.

Initial measurement data for scalar measuring components contain a single “reading” or value, while initial measurement data for interval measuring components can contain multiple readings, one for each interval that falls between the start time and stop time of the measurement.

At a simple level, initial measurement data goes through the following process:

1. Initial measurements are loaded into the system.
2. Initial measurement data is validated, edited, and estimated.
3. Initial measurements are converted into final measurements.
4. Final measurements are used to calculate usage (bill determinants, etc.).

Creating Initial Measurements

The IMD Seeder business object (D1-IMDSeeder) is used to create initial measurements (via instantiating initial measurement business objects), based on the head-end system sending the measurement. The “Initial Measurement Creation” processing method defined for the head-end system service provider defines the specific type of initial measurement to create. If for some reason an initial measurement can’t be created, an instance of the IMD Seeder business object is created to allow tracking of the error (and once the error is resolved, the IMD Seeder instance can be reprocessed).

The IMD Seeder business object determines the service provider and measuring component for the measurement (based on attributes supplied in the incoming reading, such as device ID, external reference ID, etc.). This is performed by the “Derive Service Provider and Measuring Component” pre-processing algorithm (D1-DER-SPRMC).

If the service provider is not specified on the device, the algorithm tries to derive it from the device type. Otherwise, it attempts to derive the service provider from the measurement cycle route from the service point at which the device is installed (if available).

In addition, the **Device to Service Provider Matching** field on the service provider indicates how the search for a device should be handled for initial measurements:

- **Restrict Loading to Devices Associated with Service Provider:** This is the default option when no option has been selected. Indicates that searches for devices will be based on Device Identifier, Device Identifier Type, and Service Provider.
- **Allow Loading to Any Device:** Indicates that searches for devices will be based only on Device Identifier and Device Identifier Type.

The IMD Seeder also ensures that the Start and End Date/Time fields on the initial measurement are populated, deriving them from the incoming reading if necessary. This step is performed by the “Derive IMD Date/Time Values” pre-processing algorithm (D1-VALDR-INP).

Critical Validations

The IMD Seeder also performs a series of critical validations to ensure that the incoming reading contains valid data. These critical validations include validating that the device identifier supplied is valid and exists in the system, and performing Undercount and Overcount checks for interval readings.

- **Device Identifier:** Device identifier checks validate that the device identifier provided with the measurement is valid. Device identifiers can include serial number, badge number, channel ID, etc.
- **Undercount:** An undercount occurs when an interval initial measurement contains fewer interval values than appropriate based on the interval size (or seconds-per-interval, or SPI) and the Start and End Date/Time values. For example, if an initial measurement has an interval size of one hour (or an SPI of 3600), and a Start Date/Time is October 27, 2011 and End Date/Time of October 28, 2011 (a total duration of 1 day), it should contain 24 interval values. If it contained less than 24 interval values, it could constitute an undercount.
- **Overcount:** An overcount occurs when an interval initial measurement contains more interval values than appropriate based on the interval size (or seconds-per-interval, or SPI) and the Start and End Date/Time values. For example, if an initial measurement has an interval size of one hour (or an SPI of 3600), and a Start Date/Time is October 27, 2011 and End Date/Time of October 28, 2011 (a total duration of 1 day), it should contain 24 interval values. If it contained more than 24 interval values, it could constitute an overcount.

If an incoming interval initial measurement contains either an invalid device identifier, an undercount, or an overcount, the measurement is rejected, and an instance of the IMD Seeder business object is created. Undercount and Overcount checks are performed by the "Perform Date/Time Adjustments and Undercount/Overcount Check" pre-processing algorithm (D1-DODTTMADJ).

Time Zone Translation and Conversion

If your organization receives initial measurement from a source that provides its data in a different time zone than the one in which the data will be stored in Oracle Utilities Meter Data Management, a translation can be performed to translate the time zone from an external time zone identifier to one configured within the application. This translation is defined via the Head-End Time Zone to Standard Mapping extendable lookup.

If the "Incoming Data Shift Flag" (on the device or device type) is set to "Always in Local Time", the Perform Date/Time Adjustments and Undercount/Overcount Check algorithm (D1-DODTTMADJ) converts the start and end date/time to Standard time, while taking into consideration the end of Daylight Saving Time (DST) where there is a duplicate hour.

This algorithm also performs any time zone conversion that may be necessary. If the source time zone of the incoming data differs from the target time zone then the date times will be converted to the target time zone. The source time zone is determined as follows:

1. The time zone element in the "preVEE" group,
2. The translated version of the external time zone in the "preVEE" group
3. The Service Point time zone, the Device Configuration time zone
4. The Measuring Component time zone
5. The installation time zone.

The target time zone is set to the installation time zone.

Initial Measurement Lifecycle

When initial measurements are created in the system, they move through the lifecycle defined in the initial measurement business object. The stages of this lifecycle are described below. Note that some of these states apply only “Initial Load” initial measurements.

- **Pending:** New initial measurements begin in this state. When initial measurements enter this state, Enter algorithms populate basic information for the measurement if not provided. From this state, initial measurements move on to the Additional Mapping and VEE Ready states.
- **Additional Mapping:** The Additional Mapping state allows implementations to apply custom processing to initial measurements. In the base package, initial measurements are not changed in this state.
- **Mapping Error:** Initial measurements enter this state only if errors are encountered in the Additional Mapping state. An Enter algorithm on this state creates a To Do entry.
- **VEE Ready:** When initial measurements enter this state, they are prepared for validation, editing, and estimation processing (see below for more information). an Enter algorithm on this state prepares and calculates the measurement’s “Pre-VEE” and “Post-VEE” consumption values. These values are calculated from the raw values in the initial measurement, multiplied by the Channel Multiplier/Register Multiplier (from the initial measurement’s measuring component), or Installation Constant (from the install event for the measurement’s service point).

Note: For initial measurements created from the “Manual IMD (Interval) business object (D1-ManualIMDInterval), application of multipliers to “Pre-VEE” values can be disabled via the “Apply Multiplier to Pre-VEE” business object option.

- **Error:** Initial measurements enter this state only if errors are encountered in the VEE Ready state. An Enter algorithm on this state creates a To Do entry.
- **Removed from Processing:** Initial measurements can be removed from processing from either the VEE Ready or Error states.
- **VEE Complete:** When initial measurements enter this state, validation, editing, and estimation (VEE) processing is performed. If the measurement passes the VEE processing without exceptions, it moves to the Finalized state. If an exception is encountered, it moves into the Exception state. See below for more information about VEE processing and VEE exceptions.
- **Exception:** Initial measurements enter this state if they fail one or more validation rules. An Enter algorithms on this state creates a To Do entry.
- **Discarded:** Initial measurements that generate exceptions during VEE processing can be discarded (in situations where the failed validation cannot be corrected or a replacement measurement can be obtained).
- **Force Complete:** Initial measurements that generate exceptions during VEE processing can be manually “completed” by a user. In this state, any open exceptions are closed, and the measurement can be used in downstream processing (such as usage transaction creation).
- **Finalized:** Initial measurements enter this state when they have pass VEE processing without generating exceptions. Enter algorithms on this state create Final Measurements and if necessary create Usage Transaction Correction Processor activities (in the case where a replacement measurement would impact an existing usage transaction).

Use the Business Object portal to view more detailed information about these states, including the specific algorithms executed at each state.

Validation, Editing, and Estimation

Once in the VEE Ready state, initial measurements are subject to validation, editing, and estimation. This process involves the following:

- **Validation:** Validates that the initial measurement data is within expected tolerances, and is correct
- **Editing:** If the initial measurement data is wrong in some way, the data can be automatically changed.
- **Estimation:** If initial measurement data is incomplete (for example, if one or more interval values within an interval measurement are missing), missing values can be automatically estimated.

As noted above, the values recorded in an initial measurement can change during the validation, editing, and estimation processing, and exceptions can be raised if initial measurements are wrong in some way (such as out-of-tolerance quantities, incorrect values, etc.).

Note: The validation, editing, and estimation process is referred to as VEE, and is described in more detail in a later chapter.

Skipping VEE Processing - High Quality Check for Interval Initial Measurements

Execution of VEE rules is performed by an Enter algorithm on the “VEE Complete” state of the initial measurement business object. For interval initial measurements, the “High Quality Check - Vector Band Based” (D1-HIGHQUALV) algorithm can be used to skip VEE processing and transition the initial measurement directly to the “Finalized” state.

This algorithm checks whether the intervals within an initial measurement are considered to be of “high quality” (defined below), and if they are considered “high quality”, the algorithm transitions the current initial measurement directly to the “Finalized” state.

An interval initial measurement is considered to be of “high quality” if the following conditions are met:

- There are no missing intervals in the initial measurement
- All interval values are within a range that extends above and below final interval values for the corresponding intervals from the previous reading. The range is defined by the “Low Tolerance” and “High Tolerance” algorithm parameters. These parameters define multipliers that are applied to the corresponding interval from the previous reading. The “Low Tolerance” is subtracted from the previous reading’s final measurement, and the “High Tolerance” is added to the previous reading’s final measurement to define the range within which each interval must fall.

For example, if the previous reading’s final measurement value at 1:00 AM was 20, and the “Low Tolerance” parameter is set to 0.2 and the “High Tolerance” parameter is set to 0.25, the initial measurement’s value at 1:00 AM must be between 16 ($20 - 0.2 \times 20$) and 25 ($20 + 0.25 \times 20$).

- All intervals have a condition code that falls between the range defined by the “Default Bottom Range Condition Value” and “Default Top Range Condition Value” algorithm parameters.

If the initial measurement fails any part of the high quality assessment, the routine simply exits.

Pre VEE and Post VEE Quantities

Initial measurement data contains both the original and final versions of the quantities recorded by the measuring component.

- **Pre VEE** quantities are consumption values derived from the measurements recorded by the head-end system or meter reader.
- **Post VEE** quantities are the “final” values, after VEE processing.

Pre VEE and Post VEE quantities in an initial measurement often differ based on a number of conditions, including:

1. VEE rules have changed the quantities because they are missing or obviously wrong

In this, the Pre VEE values are adjusted based on the specifics of the VEE rules applied to the initial measurement to create the Post VEE values

2. Manual changes by a user.

Condition Codes

In addition to recorded consumption values, measurements also have condition codes, used to indicate the source and quality of a measurement. For example:

- Regularly recorded measurements might have a condition code of “Regular”
- Missing measurements might have a condition code of “Missing”
- Estimated measurements might have a condition code of “External Estimated” or “System Estimated” based on where the estimation was performed.

Both Pre VEE and Post VEE values have their own condition code, which can also change during VEE processing. For example, consider the following sample measurements from an interval measuring component:

Date / Time	Pre VEE Value	Pre VEE Condition	Post VEE Value	Post VEE Condition
01/01/2010 3:00 PM	14.678	Regular	15.1	Regular
01/01/2010 4:00 PM		Missing	20	Estimated
01/01/2010 5:00 PM	13.12	Regular	13.41	Regular
01/01/2010 6:00 PM	150.12	Regular	14.12	Estimated

For the 4:00 PM interval, note the Pre VEE condition indicates the interval is missing and the Post VEE condition highlights that it was estimated.

For the 6:00 PM interval (containing a spike, or an interval with conspicuously high usage relative to surrounding intervals), note that the system head-end (the system that recorded the measurement) indicated the interval value was fine (Pre VEE is regular), but the VEE process smoothed it, and set the Post VEE condition to “Estimated.”

Condition Codes Extendable Lookup

Condition codes are defined in the Measurement Condition extendable lookup (D1-MeasurementConditionLookup). Base package condition codes delivered in this extendable lookup include the following:

Condition Code (Value)	Description
100000	No Read - System
101000	No Read - Outage
201000	Missing
301000	System Estimate
401000	External Estimate
402000	Office Estimate
501000	Regular

Condition Code (Value)	Description
901000	Super

Additional condition codes can be added to this extendable lookup to meet requirements specific to individual implementations. Gaps between the base package condition code values allow custom condition codes to be added that represent data with conditions that fall between the base package conditions. For example, a condition code to indicate that the measurement was estimated based on partially incomplete data (and is considered to be of worse quality than “System Estimate”, but better than “Missing”) would fall somewhere between 201000 and 301000.

As a general rule, condition codes used to represent bad data or error conditions should be smaller than condition codes used to represent good data.

Manually Created Initial Measurements

Initial measurements are most often created when loaded into the application from a head-end system or other external application. However, users can also manually create initial measurements via functions available from the 360 Degree View portal. The business object used for these manually create initial measurements is one of the following, depending on the type of measurement (interval or scalar):

- Manual IMD (Interval) (D1-ManualIMDInterval)
- Manual IMD (Scalar) (D1-ManualIMDScalar)

VEE Processing for Manual Initial Measurements

VEE processing for initial measurements of this type are based on the VEE Group specified in the "VEE Group For Manual Override" field on the measuring component or the "Fallback VEE Group For Manual Override" configured for the "Manual Override" VEE Group Role on the measuring component type.

Manually Editing Manual Initial Measurements

Like other initial measurements, manually created initial measurements can be manually edited via functions available in the Interval Initial Measurement Lens or Scalar Initial Measurement zones in the Initial Measurement portal. Manual edits can include changes to the measurement values or condition codes.

By default, manual edits made by users of these zones are not logged on the Log tab. Logging of manual edits to manual initial measurements can be enabled by adding a logging algorithm on an appropriate lifecycle state of the manual initial measurement business objects. The D1-LOGUSRTRN (Log User Transaction) base package algorithm can be used for this. This Enter algorithm is designed to be defined on the Initial state of the manual initial measurement business objects, but it can also be defined on any (non-transitory) Interim or Final state as well.

To ensure logging of any or all manual edits made to manual initial measurements, this algorithm should be specified on any state in which users will make manual edits. This will most often be the Pending or VEE Ready states, but could also include the Error, Exception, or Finalized states.

Note: When defining this algorithm, user should exercise caution and determine if previous algorithms in the sequence within the state contain any form of transitioning logic that may inadvertently cause this algorithm to be bypassed.

Tracking Update Dates and Times

The **Last Update Timestamp** field (LAST_UPDATE_DTTM) on the Initial Measurement Data maintenance object tracks the date and time of the last time the initial measurement was updated. This can be used to detect when a initial measurement has changed when extracting and exporting data directly from the Initial Measurement Data table. Note: This field is not displayed on the user interface.

Reader Remarks and Scalar Measurements

Initial measurements for manually-read scalar meters can create reader remarks. Reader remarks are a specific type of device event used to capture and/or record specific events or circumstances encountered when a meter reader is manually reading scalar meters.

Reader remark types are submitted with scalar initial measurements when received from a head-end system or meter read collection system. Reader remarks are NOT uploaded along with other device events. Reader remarks are ALWAYS associated with a scalar initial measurement.

Understanding Reader Remark Processing

This section provides an overview of the process that takes place when reader remarks are received within a scalar measurement. For each step in the process, the table below provides a brief description of the processing that takes place, and lists the specific objects used by the Service and Measurement Data Foundation

Note that the process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process.

Step	Process	Service and Measurement Data Foundation Objects
1.	The system receives an initial measurement from a manually-read scalar device that contains one or more reader remark types.	The base package “Initial Load IMD (Scalar)” (D1-InitialLoadIMDScalar) business object supports inclusion of one or more reader remark types.
2.	When the initial measurement enters the Finalized state, an Enter algorithm creates a reader remark for each reader remark type included in the initial measurement. Note: Creation of reader remarks does NOT make use of device event mapping extendable lookups.	Algorithm: D1-CRE-RR (Create Reader Remark)
3.	Reader remarks begin in the Pending state. They are transitioned to the Execute state either manually or via batch processing. The Execute state contains a number of Enter algorithms that perform the following <ul style="list-style-type: none"> Determine if the reader remark is eligible for processing, based on the “Eligible for Processing” flag on the reader remark type Create a To Do Entry based on the “To Do Type” and “To Do Role” defined on the reader remark type Create a service issue monitor based on the service issue monitor specified on the reader remark type Send the reader remark to subscribing systems 	Batch Control: D1-DVEVT (Device Event MO Periodic Monitor Process) Algorithm: D1-RR-ELIG (Reader Remark Eligibility) Algorithm: D1-CRE-RR-TD (Create To Do Entry for Reader Remark) Algorithm: D1-DVCEVTSIM (Create Service Issue Monitor for Reader Remark) Algorithm: D1-RRSENDSUB (Send to Subscribers for Reader Remark)

Sending Reader Remarks to Subscribing Systems

Reader remarks can be passed onto to other subscribing systems such as a customer information system (such as Oracle Utilities Customer Care and Billing), an outage system (such as Oracle Utilities Network Management System), or some other application.

Reader remarks are sent to subscribing systems when they enter the Execute state. An Enter algorithm (D1-RRSENDSUB /Send to Subscribers for Reader Remark in the base package)

determines which service providers (external systems) are interested in receiving the current reader remark based upon its type and category, finds the processing method for each, and processes the device event accordingly.

The means of sending reader remark information to subscribing systems is defined in the “How to Send Device Event Related Information” processing method for the service provider representing the subscribing system. Reader remark information can be sent via outbound communication business object, outbound communication, or batch process.

The method for sending device events to subscribing systems (business object, outbound message, or batch process) is defined for the “Reader Remark” device event category, and can be overridden for individual reader remark types (including the ability to exclude specific reader remark types. In addition, a default event processing method can be also be configured that applies when methods of transmission aren't specified at the individual category level.

The “Subscribe to Device Event” business service is used to process reader remark subscription requests, and allow external applications to manage the categories of events they receive.

Reader Remarks and Service Issue Monitors

Like other device events, reader remarks can create service issue monitors. See **Service Issue Monitors and Service Investigative Orders** in **Chapter 8: Device Communication and Device Events** for more information about service issue monitors.

Subtractive Measuring Components

Initial measurement data for subtractive measuring components (such as most scalar measuring components) also typically contain start and stop readings in addition to Pre and Post VEE usage. For example, a set of initial measurements for a subtractive scalar measuring component might look like the following:

Date / Time	Start Reading	Stop Reading	Pre VEE Usage	Pre VEE Condition	Post VEE Value	Post VEE Usage
01/01/2010 3:00 PM	0	1500	1500	Regular	1515	Regular
02/2/2010 4:11 PM	1500	2100	600	Regular	606	Regular
03/03/2010 5:22 PM	2100	2900	800	Regular	808	Regular
04/01/2010 1:00 PM	2900	3500	600	Regular	606	Regular

Rollover Calculations

Subtractive measuring components can “rollover” when the reading exceeds the maximum value based on the number of dials. For example, a register with a 4 dials can record values up to 9999 before rolling over to 0000. When this occurs, consumption is calculated based on the following attributes and calculated values.

- **Rollover Threshold** is the percentage of the measuring component's dial capacity at which measurements for measuring components of this type are considered to have rolled over. Dial capacity is the largest value that can be recorded for the measuring component, based on the measuring component's number of dials. For example, a measuring component with 5 dials has a dial capacity of 99999.
- **MaxDialCapacity** is the maximum value for the number of dials, rounded up to the next whole multiple of 10 (or 10 raised to the power of the number of dials). For example, for a register with 4 dials, the MaxDialValue is 10000.
- **MaxAcceptableDifference** is the maximum acceptable consumption that can be recorded for the register. This is equal to the MaxDialCapacity multiplied by the rollover threshold. For example, the MaxAcceptableDifference for a register with 4 dials and a rollover threshold of

90% would be 9000. If the consumption is greater than this value, the initial measurement is transitioned to the Error state.

- **Difference:** The difference between the Stop Reading and Start Reading, obtained by subtracting the Start Reading from the Stop Reading. If the Difference is less than zero (<0), then add the MaxDialCapacity to calculate Rollover.
- **Rollover:** The adjusted consumption for a reading on a register that has rolled over. Only applicable if the Difference (Stop Reading - Start Reading) is less than zero (<0).
- **Consumption:** The calculated consumption for the reading, equal to either the Difference or Rollover. If the Difference is greater than or equal to zero, consumption is equal to the Difference. If the Difference is less than zero (<0), and the Rollover is less than or equal to the MaxAcceptableDifference, the consumption is equal to the Rollover.

Example: Consider an initial measurement with the following attributes:

- Number of Dials: 4
- Rollover Threshold: 90 (%)
- Start Reading: 8900
- Stop Reading: 0500

For this reading,

MaxDialCapacity = 10000

MaxAcceptableDifference = 9000 (10000 * 90)

Difference = 0500 (Stop Reading) - 8900 (Start Reading) or **-8400**

Rollover = 10000 (MaxDialCapacity) + -8400 (Difference) or **1600**

Consumption is equal to **1600** (Rollover).

Final Measurements

When an initial measurement is considered “final,” that is, it has passed all VEE processing and no additional modifications or changes need to be made, it is transformed into a Final Measurement, or simply a Measurement (the terms measurement, final measurement, and final consumption all reference this same “final” measurement data).

When creating final measurements from initial measurement data:

- Final measurements are created using Post VEE quantities
- Each final measurement's condition is copied from the Post VEE condition
- Initial measurements are normalized into final measurements where each final measurement is for a specific date and time.
- Because a single initial measurement may contain many “readings,” a separate final measurement is created for each interval in the initial measurement. For example, if an initial measurement contains 24 hours of 15 minute readings, 96 measurements will be created, each with a specific date and time.

Attributes used to define final measurements include the following:

- **Measuring Component:** The measuring component which recorded the initial measurement from which the final measurement was derived.
- **Measurement Date/Time:** The date and time of the final measurement.
- **Condition:** The condition code of the final measurement.
- **Measurement Use:** A flag that indicates if the measurement should be used when calculating usage. "Do Not Use" indicates that the measurement has been replaced and

should not be used in usage calculations. This flag is set when replacement usage received that overlaps existing measurements is finalized. If an existing measurement falls within the time period of the incoming initial measurement for the same measuring component, the existing measurement is flagged as "Do Not Use".

- **User Edited:** A flag that indicates if the measurement has been manually edited by a user.
- **Initial Measurement:** The initial measurement from which the final measurement was derived.
- **Previous Measurement Date/Time:** The date and time of the previous measurement for the same measuring component.
- **Measurement:** The measurement value, based on Value Identifiers configured for the measuring component's type.
- **Value 1 (2-10):** Other values derived from the measurement, based on Value Identifiers configured for the measuring component's type.
- **Local Date/Time:** The local date and time for the measurement (after date/time timezone conversions).
- **Last Update Timestamp (LAST_UPDATE_DTTM):** The date and time of the last time the measurement was updated. This can be used to detect when a measurement has changed when extracting and exporting data directly from the Measurement table. Note: This field is not displayed on the user interface.

Final measurements are periodically transformed into more concise and accessible usage (also known as bill determinants) for the subscribing systems. In this example, a time-of-use map is applied to the final measurements for an entire month. The usage calculation process is described in more detail in a later chapter.

Derived Values

Final measurements can record up to 10 derived values in addition to the "as measured" value. The derivation formula for each value on a final measurement is held in an algorithm and therefore can derive anything. For example, a set of measurements can adjusted or converted into other units of measure:

Date / Time	As Measured UOM: CCF SQL: n/a	Loss Adjusted UOM: CCF SQL: n/a	Thermal Units UOM: BTU SQL: n/a
01/01/2010 3:00 PM	10	10.1	10.11
01/01/2010 4:00 PM	15	15.15	15.165
01/01/2010 5:00 PM	10	10.1	10.11

Derived values are not reliant on consumption values, but can also come from factors, historical data, or another source. For example, measurements might be compared to "normal" usage for the usage period:

Date / Time	As Measured UOM: CCF SQL: n/a	Loss Adjusted UOM: CCF SQL: n/a	Thermal Units UOM: BTU SQL: n/a	Normal CCF UOM: CCF SQL: Normal	Percent (%) of Normal UOM: CCF SQL: n/a
01/01/2010 3:00 PM	10	10.1	10.11	10	100%
01/01/2010 4:00 PM	15	15.15	15.165	10	150%
01/01/2010 5:00 PM	10	10.1	10.11	10	100%

Updating Final Measurements

Final measurements can be updated if necessary. If final measurements are discovered to be incorrect (for whatever reason), **a new initial measurement is created to correct them**. This new initial measurement contains the corrected consumption, and after the initial measurement has completed VEE processing, the existing final measurements are updated with the newly calculated consumption.

Note: The primary key on the table used to store final measurements (the Measurement table) is a combination of the measuring component ID and the date/time of the measurement. This means that it is impossible for more than one final measurement to exist for a measuring component for a given date/time.

The reason a new initial measurement must be created and completed to add or update measurements is because there is no user interface that allows users to directly edit final measurements.

Daylight Saving Time Support

This section describes how the Oracle Utilities Service and Measurement Data Foundation and its related products support Daylight Saving Time (DST) for measurement data, including:

- **Types of Devices**
- **Date/Time Storage and Display**
- **Oracle Utilities Application Framework**
- **Typical Daylight Saving Time Scenarios**

Types of Devices

In Oracle Utilities Service and Measurement Data Foundation initial measurement data processing, the application understands a device that is either:

- a. Aware of the fact that Local time in the device's time zone has been shifted from "Standard", or
- b. Unaware of any such shifting.

Devices in the "unaware" category ("b") will always send Oracle Utilities Service and Measurement Data Foundation initial measurement data with measurements in Standard time. Devices in the "aware" category ("a") will always send the application initial measurement data in Local time.

Whether a device falls into category "a" (Aware) or "b" (Unaware) is configured via the **Incoming Data Shift** flag on the device type (which can be overridden on the device). The values of the flag are:

- "Always in Local Time" (used with "aware" devices, or category "a")
- "Always in Standard Time" (used with "unaware" devices, or category "b")

This flag is used by pre-processing algorithms (Perform Date/Time Adjustments and Undercount/Overcount Check) in the IMD Seeder business object to convert any date/times on the initial measurement into standard time. Note that this conversion is only done if the device falls into category "a."

Date/Time Storage and Display

Within the database, measurements are stored with two (2) date/times: Standard and Local. The Service and Measurement Data Foundation uses the date/time in Standard as part of the prime key of the measurement table. The presence of the Local date/time field facilitates querying measurement data using local time.

When displaying dates and times for initial measurement data:

- Display of the data on the Oracle Utilities Meter Data Management **360 View** is in Local time.
- The **IMD Lens** zone (in the Oracle Utilities Meter Data Management version of the Initial Measurement portal) also displays data in Local time.
- The **Raw Data, Pre-VEE and Post-VEE XML Data** zone on the Initial Measurement portal does not shift the data into Local time, so if that the pre-processing algorithm has shifted the data into standard time, the date/times displayed will be in Standard time. Please note that the only two date/times visible in that zone will typically be the Start date/time and End date/time of the initial measurement; the Service and Measurement Data Foundation strips off the date/times from the individual intervals of the initial measurement at pre-processing time.
- The **Measurement** zone shows both the local and standard date/times as-is.

Oracle Utilities Application Framework

When during initial measurement data processing, it is determined that time shifting is required, the meter data management looks at the time zone metadata in the application. Oracle Utilities Application Framework utilizes the configuration of an Olson DB time zone code on the time zone metadata. This Olson DB contains the shift date/times for every time zone across the globe.

In North America for example, the available Olson DB time zone codes are much more specific than "Eastern/Central/Mountain/Pacific", and include details for areas places such as Arizona and Indiana where there may or may not be shifting for daylight saving time.

Oracle Utilities Application Framework provides business services that wrap the application services that perform time shifting. These services use the time zone metadata to retrieve shift date/times using the Olson DB.

Typical Daylight Saving Time Scenarios

The following table illustrates typical daylight saving time scenarios.

Time Springs Forward		Other Days		Time Falls Back	
DST Shifted Meter in Local Time	Shift & Store time as standard in IMD	DST Shifted Meter in Local Time	Shift & Store time as standard in IMD	DST Shifted Meter in Local Time	Shift & Store time as standard in IMD
3/14/2011	3/14/2011	7/18/2011	7/18/2011	11/7/2011	11/7/2011
1:00	1:00	1:00	0:00	1:00	0:00
3:00	2:00	2:00	1:00	2:00	1:00
4:00	3:00	3:00	2:00	2:00	2:00
5:00	4:00	4:00	3:00	3:00	3:00
6:00	5:00	5:00	4:00	4:00	4:00
7:00	6:00	6:00	5:00	5:00	5:00
8:00	7:00	7:00	6:00	6:00	6:00
9:00	8:00	8:00	7:00	7:00	7:00
10:00	9:00	9:00	8:00	8:00	8:00
11:00	10:00	10:00	9:00	9:00	9:00
12:00	11:00	11:00	10:00	10:00	10:00
13:00	12:00	12:00	11:00	11:00	11:00
14:00	13:00	13:00	12:00	12:00	12:00
15:00	14:00	14:00	13:00	13:00	13:00
16:00	15:00	15:00	14:00	14:00	14:00
17:00	16:00	16:00	15:00	15:00	15:00
18:00	17:00	17:00	16:00	16:00	16:00
19:00	18:00	18:00	17:00	17:00	17:00
20:00	19:00	19:00	18:00	18:00	18:00

Time Springs Forward		Other Days		Time Falls Back	
DST Shifted Meter in Local Time	Shift & Store time as standard in IMD	DST Shifted Meter in Local Time	Shift & Store time as standard in IMD	DST Shifted Meter in Local Time	Shift & Store time as standard in IMD
21:00	20:00	20:00	19:00	19:00	19:00
22:00	21:00	21:00	20:00	20:00	20:00
23:00	22:00	22:00	21:00	21:00	21:00
0:00	23:00	23:00	22:00	22:00	22:00
		0:00	23:00	23:00	23:00
				0:00	0:00
23 hours	23 hours	24 hours	24 hours	25 hours	25 hours

Bold-faced entries indicate times that are impacted by daylight saving time conversion.

Understanding Estimated Initial Measurements

This section provides an overview of estimated initial measurements and how they are used in Oracle Utilities Service and Measurement Data Foundation and Oracle Utilities Meter Data Management.

Estimated Initial Measurements

Over the course of time, it may happen that the system will not receive usage for a device for some period of time. When the system detects that a measuring component is missing final measurements, it can create an initial measurement via estimation rules. This type of initial measurement is referred to as an estimated initial measurement (as opposed to an Initial Load or Manual initial measurement).

At a high-level, the estimation process is as follows:

- Missing Measurements are detected by a “Periodic Estimation” system event algorithm on the measuring component business object
- Estimated initial measurements are created for the “missing” time period by the “Period Estimation” system event algorithm on the measuring component business object
- Values and consumption for the estimated initial measurements are calculated by “estimation” VEE rules.

It’s important to note that the processes that detect missing measurements do NOT themselves estimate consumption. Rather, these detection processes simply create an initial measurement and let the estimation VEE rules estimate the consumption for the initial measurement.

Detecting Missing Measurements

The detection of missing measurements can occur at different points in time for interval and scalar measuring components:

- A batch process exists to initiate creation of estimated initial measurements.
- For manually read scalar measuring components, a usage calculation rule can create initial measurements if it cannot find a measurement and it has been given permission to estimate.

Interval Measuring Components

The batch process that detects missing consumption for interval measuring components uses two elements on the “Interval Channel Type - Physical” (D1-IntervalChannelTypePhysical) measuring component type:

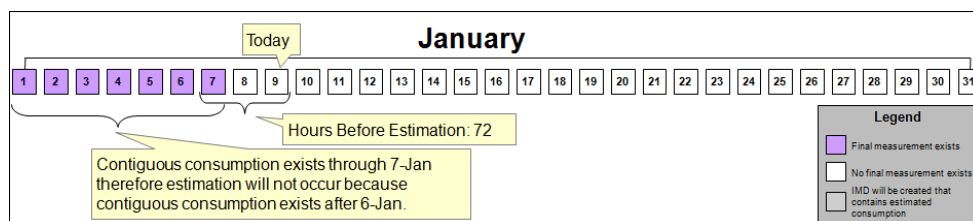
- **Hours Before Estimation:** the number of hours after the End Date and Time of the most recent measurement that must pass before the measuring component is considered due for estimation. For example, if set to 72 hours, estimation will only take place 72 hours after the End Date and Time of the latest measurement.
- **Number of Hours to Estimate:** the number of hours of measurement data that are estimated when estimation is performed for the measuring component.

If a measuring component has contiguous final measurements on/after a date and time equal to the current date/time minus the “Hours Before Estimate” value, the measuring component is not estimated.

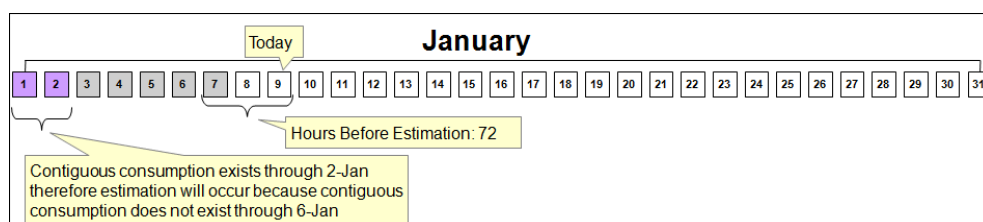
If the measuring component does NOT have contiguous final measurements on/after a date and time equal to the current date/time minus the “Hours Before Estimate” value, the measuring component is estimated. If the measuring component is subject to estimation, measurements will be estimated through the a date and time equal to the current date/time minus the (Hours Before Estimation - Number of Hours to Estimate).

The following examples illustrate how the system determines if estimation should take place. These examples are all assume a measuring component whose measuring component type specifies an “Hours Before Estimation” of 72 (3 days) and a “Number of Hours to Estimate” of 24 (1 day).

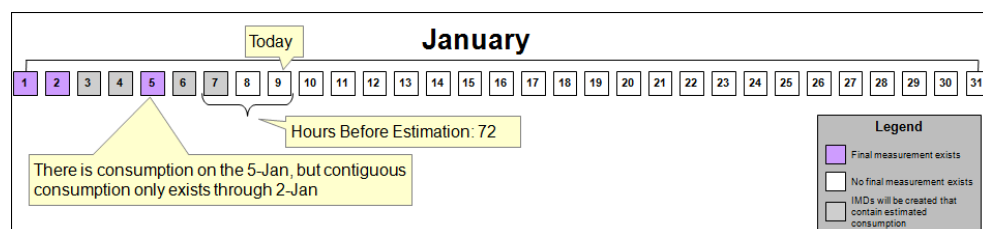
Example 1: On January 9, if there is contiguous consumption through January 7, estimation would not occur because consumption exists after January 6 (3 days prior to January 9). The diagram below illustrates this example:



Example 2: On January 9 if contiguous consumption existed through January 2, estimation would take place because consumption does not exist through January 6 (3 days prior to January 9). In this example, consumption would be estimated for missing intervals from January 2 through January 7 (January 9 minus 48 hours (Hours Before Estimation minus Number of Hours to Estimate)). Note that this assumes that period estimation had NOT been run after January 5. The diagram below illustrates this example:



Example 3: On January 9 if contiguous consumption existed through January 2 and for January 5 (but NOT for January 3-4 or 6-7), two initial measurements would be created, one for January 3-4, and another for January 6-7. The diagram below illustrates this example:



Scalar Measuring Components

The batch process that detects missing consumption for scalar measuring components uses elements on the “Auto-Read Register Type” (D1-AutoReadRegisterType) measuring component type:

- **Expected Hours Between Measurements:** This indicates the number of hours that will separate each measurement.
 - **6:** When 6 is selected there would be 4 expected measurements each day: 12:00:00AM, 06:00:00AM, 12:00:00PM, and 06:00:00PM.
 - **8:** When 8 is selected there would be 3 expected measurements each day: 12:00:00AM, 08:00:00AM, and 04:00:00PM.
 - **12:** When 12 is selected there would be 2 expected measurements each day: 12:00:00AM, and 12:00:00PM.

- **24:** When 24 is selected there would be 1 expected measurement each day: 12:00:00AM.

Note: the above expected measurement times all assume a **First Daily Measurement** of 12:00:00AM.

- **First Daily Measurement Time:** This indicates the time that the first measurement of the day is expected to arrive. Each subsequent expected measurement is then adjusted to maintain a consistent number of hours between each measurement. For example, if the **Expected Hours Between Measurements** was set to 8 and the **First Daily Measurement Time** was set to 02:00:00AM the expected reads each day would be: 02:00:00AM, 10:00:00AM, and 06:00:00PM. This must be set to a value that is less than the **Expected Hours Between Measurements**. For example, if the **Expected Hours Between Measurements** was 8 the latest this field could be set to is 07:59:59.
- **Early Measurement Threshold:** This threshold is used to determine whether a measurement that was received early satisfies the next expected measurement date/time. For example, if the threshold was set to 02:00:00 and a measurement was received at 07:45:00AM it would satisfy an expected measurement of 08:00:00AM. If no measurement should be considered early the threshold should be set to 00:00:00.
- **Late Measurement Threshold:** This threshold is used to determine whether a measurement that was received late is too late to satisfy any of the expected measurement date/times. For example, if the threshold was set to 02:00:00 and a measurement was received at 05:45:00AM it would not satisfy the expected measurement of 12:00:00AM or 08:00:00AM. If no measurement should be considered late the threshold should be set to one second less than the **Expected Hours Between Measurements**. For example, if the **Expected Hours Per Measurement** was 8, the late threshold should be set to 07:59:59 if no measurements should be considered late.
- **Hours Before Estimation:** This is the number of hours after the **End Date and Time** of the most recent measurement that must pass before the measuring component is considered due for estimation. For example, if it were set to 72 hours and the process date time of the periodic estimation process was 01/05/2013 12:00:00AM, only those measuring components with a **Latest Measurement Date Time** of 01/01/2013 11:59:59AM or older would be evaluated for estimation. If this is left blank or set to zero, periodic estimation will not be executed for any measuring components of this type.
- **Number of Hours to Estimate:** This specifies the minimum number of hours of measurement data that should be estimated when periodic estimation is executed for measuring components of this type. If this is left blank or set to zero, periodic estimation will not be executed for any measuring components of this type.

When the periodic estimation process is executed for auto-read registers, it searches for gaps in the final measurements by first aligning all the final measurements to one of the expected measurement date/times then comparing that list of final measurements against the full schedule of expected measurement date/times. For any expected measurement date/times that do not have an associated final measurement or an in process initial measurement an estimate initial measurement will be created. The “Auto-Read Scalar Periodic Estimation” algorithm type (D1-ARSPE) is used to create estimation initial measurements for auto-read registers. Use the Algorithm Type portal to view more details about this algorithm type.

The **Hours Between Measurements** and the **First Daily Measurement Time** fields are used to determine the schedule of expected measurement date/times for auto-read registers. For example, if there were 6 hours between measurements and the first measurement time was 12:00:00AM then each day readings would be expected at the following times: 12:00:00AM, 06:00:00AM, 12:00:00PM, and 06:00:00PM.

The **Early Threshold** and **Late Threshold** fields are used to determine if a measurement that was received should satisfy one of the expected measurement date/times. This is to accommodate scalar registers that do not always produce measurements which fall on an exact and consistent time. For example the **Early Threshold** can be used to allow measurement received at

11:50:00PM to satisfy the 12:00:00AM expected measurement while the **Late Threshold** can be used to prevent that same 11:50:00AM measurement from satisfying the 06:00:00PM expected measurement.

The **Latest Measurement Date/Time** and **Adjusted Latest Measurement Date/Time** fields are maintained on auto-read measuring components. The **Latest Measurement Date/Time** will be the exact measurement date time and the **Adjusted Latest Measurement Date/Time** will be the measurement date time aligned to one of the expected measurement date/times based upon the measuring component type configuration (specifically the **Expected Hours Between Measurements, First Daily Measurement Time, Early Threshold**, and **Late Threshold** fields). The “Update Latest Date/Time on Scalar Measuring Component” algorithm type (D1-UPDDTSCMC) is used to update the latest date/time for scalar measuring components. Use the Algorithm Type portal to view more details about this algorithm type.

Note: If periodic estimation is being implemented for auto-read scalar devices that have existing measurement history, the "No of Hours in Past to Retrieve Last Usable Measurement" option of the "Measurement Data Option" Feature Configuration type should be configured (see **Measurement Data Options** on page 2-3 for more information). When this option is set, the latest measurement information on the measuring component, when not present, will be initialized using the most recent usable final measurement that is found for the measuring component. This will be done with the first execution of the periodic estimation process, after the first execution this feature configuration should be removed.

Estimation Algorithms

Estimation for measuring components is detected and initiated by two algorithms, one on the device business object, and the other on the measuring component business object.

Devices whose measuring components are subject to periodic estimation have a “Periodic Estimation” Monitor algorithm (D1-PERESTM) on their Active state. This monitoring algorithm retrieves the device's measuring component and invokes the algorithm on the “Periodic Estimation” system event on the measuring component business object. Note: the device monitoring algorithm can be configured to look at device configurations from a number of days in the past defined by an algorithm parameter. The base package includes this algorithm on the Active state of the Smart Meter (D1-SmartMeter), Manual Meter (D1-ManualMeter), and Communication Component Device (D1-CommComponentDevice) device business objects. The “Periodic Estimation” monitor algorithm can be triggered by the “Smart Meter State Monitor Process” batch process (D1-SMMTR).

The “Periodic Estimation” System Event algorithms determine if a measuring component is missing final measurements (as described above), and if final measurements are missing, estimates initial measurements and/or creates a To Do Entry. In the base package:

- The algorithm for Interval Channel (D1-IntervalChannel) measuring components is the “Create Interval IDM and To Do Based Upon Install History” algorithm (D1-CIITBIH). Algorithm parameters can specify if an initial measurement is to be created, if an To Do Entry is to be created, and the To Do Type and Role (if applicable) for any To Do entries created.
- The algorithm for Register - Auto-Read (D1-RegisterAutoRead) measuring components is the “Auto-Read Scalar Periodic Estimation” algorithm (D1-ARSPE). Algorithm parameters can specify if an initial measurement is to be created, if an To Do Entry is to be created, and the To Do Type and Role (if applicable) for any To Do entries created.

Estimation Calculations

When new estimated initial measurements are created (for either interval or scalar measuring components), the Pre-VEE and Post-VEE values are initially set to zero. When the estimate initial measurement enters the “VEE Complete” state, the VEE rules within the VEE group defined for the “Estimation” VEE role (defined by the “VEE Group For Estimation” field on the measuring component) calculate values and consumption for the initial measurement.

Note that the VEE rules used in this process can also include validation rules that perform validation on the estimated values and consumption. For example, the “estimation” VEE group might contain rules to estimate interval values from a profile, and then validate the resulting interval measurement against measurements from a consumption reference measuring component. In this example, the validation would be configured to be applied after the interval profile estimation.

Estimated initial measurements are created using “estimation” initial measurement business objects. The base package includes the Estimation IMD (Interval) and Estimation IMD (Scalar) business objects (D1-EstimationIMDInterval or D1-EstimationIMDScalar, respectively).

Related Measuring Component Consumption Synchronization

This section describes how consumption for related consumption channels can be kept synchronized using the "Related Measuring Component Consumption Synchronization" process. This includes:

- **Overview**
- **Understanding Consumption Synchronization Processing**
- **Consumption Synchronization Configuration Settings**
- **Configuration Example**
- **Consumption Synchronization Examples**

Overview

Traditional interval and smart meters are often configured with two channels of consumption data, one interval and the other scalar. Utilities configure their meters this way for several reasons, including:

- Two channels provide for a redundant reading in case of a failure
- Two channels provide for an independent check reading to ensure the meter is working properly
- Customers often need both types of data for their business processes.

When capturing and storing usage for smart meters with related measuring components, one potential challenge is keeping measurements from scalar and interval channels synchronized. Measurements for interval and scalar channels for the same device can vary for many different reasons, such as:

1. Some meters use different technologies for recording interval and register readings, so the physical readings can be different
2. Readings can vary because of different measurement times
3. Estimation processes can vary by channel and produce different results
4. There can be a mix of actual and estimated measurements

Having two sources of consumption data can cause many problems for utilities. Unfortunately, utilities often need both the scalar and interval data to satisfy their business requirements. For example, scalar register readings may be used for billing, while the interval data is used on a customer self-service portal.

Consumption Synchronization

Consumption synchronization is a process by which measurement data for related measuring component is kept synchronized by re-estimating existing measurements when measurement data for related measuring components is received.

For example, consider the following scenario:

- A meter has both scalar and interval channels, and the scalar register channel is considered the "primary" or more reliable channel. The total consumption for each channel should match the consumption for the other channel.
- For a particular day, a daily reading for the "secondary" (interval) channel is received before reads for the scalar register have been received. This reading contains two missing intervals, which are estimated upon receipt into the system.

- After the interval reading has been finalized, a scalar register reading for the same day is received. The total of this reading is slightly different than the total of the interval reading due to the estimation of the missing intervals.
- The consumption synchronization process can trigger a re-estimation of the two estimated intervals to adjust the total of the interval reading to match that of the scalar register reading.

Other examples of situations that can be resolved using the consumption synchronization process include the following:

- Register Readings whose start/stop times do not align with interval boundaries
- A broken meter for which estimated readings are created for several days
- Requirements that readings from both channels match exactly even if neither contains estimated measurements

Primary and Secondary Measuring Components

When measuring components are related, one is considered the "primary" measuring component, and the other is considered the "secondary" measuring component. How the consumption synchronization process works for a given meter is based largely on which measuring component is considered primary and which is considered secondary.

The following are important principles regarding the primary/secondary measuring components:

1. The primary measuring component has the more reliable data.
2. The secondary measuring component should validate that its consumption aligns with the primary measuring component.
3. If the secondary measuring component needs to be estimated, it should estimate using the available data from the primary measuring component.
4. When possible, initial measurements for the primary measuring component should be processed prior to those for the secondary measuring component (Note: the D1-IMD batch process will process initial measurements in this order).*
5. Periodic estimation should be performed on the primary measuring component first, and then the secondary measuring component.

* In some scenarios when the secondary measuring component is read more frequently than the primary measuring component it can be advantageous to violate this principle in favor of being able to execute smaller more frequent loads of measurement data.

See **Measuring Components - Primary and Secondary Measuring Components** on page 6-27 for more information about how to specify which measuring component is primary.

Understanding Consumption Synchronization Processing

This section provides an overview of the process that supports keeping consumption for related measuring components synchronized.

Note that the process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process.

Step	Process	Service and Measurement Data Foundation Objects
1.	An initial measurement is received (via initial load or manual) for a measuring component whose device is configured to use consumption synchronization.	Consumption synchronization is enabled via the "Keep Consumption Reference MC in Sync" parameter on the device's device configuration type. Device Configuration Type Business Object: D1-DeviceConfigurationType
2.	Consumption synchronization is triggered if measurements for a "consumption reference measuring component" have condition codes that fall within the "consumption synchronization" range. The "Update Latest Measurement Date/Time" Enter algorithm on the "Finalized" state of the initial load and manual initial measurement business objects evaluate measurements on the related measuring component. If this algorithm finds measurements that fall within the time span of the incoming initial measurement and that have a measurement condition within the range defined for consumption synchronization, it creates a consumption synchronization activity. The specific activity (interval or scalar) created is based on the type of consumption measured by the consumption reference measuring component. Activities are created in the "Pending" state.	Initial Load/Manual Interval Algorithm: D1-UPD-DTMC (Update Latest Measurement Date/Time on MC with Consumption Sync) Initial Load/Manual Scalar Algorithm: D1-UDTSCMCRE (Update Latest Measurement Date/Time on Scalar MC with Consumption Sync) Interval Consumption Synchronization Activity Business Object: D1-RelMCREInterval Scalar Consumption Synchronization Activity Business Object: D1-RelMCREScalar

Step	Process	Service and Measurement Data Foundation Objects
3.	<p>The "Related MC Consumption Sync" batch process transitions the consumption synchronization activity from "Pending" to the "Re-Estimate" state.</p> <p>The "Consumption Sync" Enter algorithm on the "Re-Estimate" state creates estimation initial measurements.</p> <ul style="list-style-type: none"> For scalar measuring components, this initial measurement contains no measurement value. For interval measuring components, this initial measurement can contain a combination of "Regular" measurements (based on existing final measurements) and measurements that require estimation (based on and corresponding to the measurements with condition codes that fell between the range defined for "consumption synchronization" in step 2). 	<p>Related MC Consumption Sync Batch Control: D1-RMCRE</p> <p>Interval MC Consumption Sync Algorithm: D1-RE-INTVMC (Interval MC Consumption Sync)</p> <p>Scalar MC Consumption Sync Algorithm: D1-RE-SCMC (Scalar MC Consumption Sync)</p> <p>Estimation Initial Measurement (Interval) Business Object: D1-EstimationIMDInterval</p> <p>Estimation Initial Measurement (Scalar) Business Object: D1-EstimationIMDScalar</p>
4.	<p>The "Perform VEE for Estimation IMD" Enter algorithm the "VEE Complete" state of the estimation initial measurement business objects perform estimation calculations for measurements that require estimation, using VEE rules defined in the "Estimation VEE Group" for the measuring component.</p>	<p>Estimation Algorithm: D1-ESTM-VEE (Perform VEE for Estimation IMD)</p>

The following sections provide addition detail for the various configuration options and settings used in performing consumption synchronization.

Consumption Synchronization Configuration Settings

This section describes the various configuration settings and options that can be used to configure consumption synchronization for related measuring components. This includes:

- **Device Configuration Type**
- **Measuring Component Type**
- **Measuring Components - Primary and Secondary Measuring Components**
- **Initial Load and Manual Initial Measurement Algorithms**
- **Consumption Synchronization Activities and Algorithms**
- **Consumption Synchronization Activities and Algorithms**
- **VEE Rules**

Device Configuration Type

Device configuration types are used to define which related measuring components are to be kept synchronized. The table below outlines the device configuration type parameters used by the consumption synchronization process.

Parameter	Description
Keep Consumption Reference MC in Sync	<p>Defines if consumption reference measuring components should be kept in sync with their primary measuring component. Valid options include:</p> <ul style="list-style-type: none"> • No: This indicates that the related measuring components should not be kept in sync (leaving this field blank will yield the same result) • Secondary Only: This indicates that the related measuring components consumption synchronization will only be triggered for the secondary measuring component when new measurement data is received for the Primary measuring component. • Primary and Secondary: This indicates that the related measuring component consumption synchronization can be triggered by new measurement data from either the primary measuring component or the secondary measuring component. When this value is selected, the Minimum Condition to Sync Primary MC field is required.

Parameter	Description
Minimum Condition to Sync Primary MC	<p>This controls the consumption synchronization of the primary measuring component based when measurements are received for the secondary measuring component.</p> <p>For example, if the interval measuring component was primary and the secondary scalar measuring component received a measurement for a period where there were existing interval measurements that were eligible for consumption synchronization and this field were set to 'Regular' (501000) then the scalar measurement would need a condition that was equal to or greater than 'Regular' (501000) to trigger a consumption synchronization of the interval measurements.</p> <p>This field is only required when the Keep Consumption Reference MC in Sync field is set to 'Primary and Secondary'.</p>
Sum Check VEE Exception Type	<p>Defines a VEE exception type that the system will check for to prior evaluating if consumption synchronization is necessary. If no VEE exceptions of this type are found, that indicates the primary and secondary measuring components' consumption are within tolerance of each other and therefore there is no need for consumption synchronization.</p>

Measuring Component Type

Scalar registers used with the consumption synchronization process must be created based on the "Register - Auto-Read" measuring component (D1-RegisterAutoRead) and "Auto-Read Register Type" measuring component type (D1-AutoReadRegisterType) business objects. The table below outlines the "Auto-Read Register Type" measuring component type parameters used by the consumption synchronization process.

Parameter	Description
Ignore Estimates as IMD Start Reading	<p>Specifies if any existing measurements should be retained when a non-estimated measurement is received. Options include:</p> <p>No: Estimates will be used as a start reading so long as there are no roll-over threshold errors. Note: this is the default behavior if no value is selected.</p> <p>Yes: If a non-estimated initial measurement is received, any estimated measurements prior to the initial measurement being processed will be flagged as "Do Not Use." The start reading for the non-estimated initial measurement will be the first non-estimated reading found.</p>

Parameter	Description
Flag Future Estimates as Do Not Use	<p>Controls the handling of estimates that exist chronologically after the non-estimated initial measurement being processed. Options include:</p> <p>No: No estimates will be flagged as "Do Not Use." Note: this is the default behavior if no value is selected</p> <p>Yes: Any estimates after the current initial measurement until the first non-estimate will be flagged as "Do Not Use."</p> <p>Yes (Only if there is a Next Actual): Any estimates after the current initial measurement until the first non-estimate is found will be flagged as "Do Not Use." If no non-estimates are found, no future estimates will be flagged as "Do Not Use."</p>
Actuals or Corrections Initiate Re-Estimation	<p>Controls whether or not any estimations that are flagged as "Do Not Use" during the initial measurement finalization process should be re-estimated. Options include:</p> <p>No: None of the estimates flagged as "Do Not Use" will be re-estimated. Note: this is the default behavior if no value is provided.</p> <p>Yes: Any gaps created by estimates that were flagged as "Do Not Use" will be re-estimated by the periodic scalar estimation process. This is done by resetting the Latest Read Date/Time and Adjusted Latest Read Date/Time on the measuring component.</p>

Measuring Components - Primary and Secondary Measuring Components

The relationship between the primary and secondary measuring components is specified using the "Consumption Reference Measuring Component" parameter on the measuring component.

A measuring component is considered to be secondary when it holds the relationship to the Consumption Check Measuring Component (the primary measuring component).

For example, consider the following related measuring components.

- Scalar Measuring Component: ER-SM-007 / 1 / Electric kWh Daily
- Interval Measuring Component: ER-SM-007 / 2 / Electric kWh 60min

If the scalar measuring component is the primary measuring component, it does NOT specify a "Consumption Reference Measuring Component", and the interval measuring component specifies the scalar measuring component as the "Consumption Reference Measuring Component".

If the interval measuring component is the primary measuring component, it does NOT specify a "Consumption Reference Measuring Component", and the scalar measuring component specifies the interval measuring component as the "Consumption Reference Measuring Component".

Remember that pending initial measurements for the secondary measuring component are processed after the primary measuring component's initial measurements. The D1-IMD batch process will process initial measurements in this order

Initial Load and Manual Initial Measurement Algorithms

The "Update Latest Measurement Date/Time" algorithms on the Finalized state of the "Initial Load" and "Manual" initial measurement business objects determine whether or not consumption synchronization is required by evaluating measurements on the in initial measurement's related measuring component.

If this algorithm finds measurements that fall within the time span of the incoming initial measurement and that have a measurement condition within the range defined for consumption synchronization (see parameters below), it creates a consumption synchronization activity. The specific activity (interval or scalar) created is based on the type of consumption measured by the consumption reference measuring component. Activities are created in the "Pending" state.

The table below lists the specific algorithms used by the interval and scalar initial measurement business objects.

Initial Measurement	Algorithm
Initial Load IMD (Interval) (D1-InitialLoadIMDInterval)	D1-UPD-DTMC (Update Latest Measurement Date/Time on MC with Consumption Sync)
Manual IMD (Interval) (D1-ManualIMDInterval)	
Initial Load IMD (Scalar) (D1-InitialLoadIMDScalar)	D1-UDTSCMCRE (Update Latest Measurement Date/Time on Scalar MC with Consumption Sync)
Manual IMD (Scalar) (D1-ManualIMDScalar)	

These algorithms all use the same set of parameters:

- **Interval Consumption Sync Activity:** Defines the business object for the activity to be created when the related measuring component is interval. In the base package, this is set to "D1-RelMCREInterval".
- **Scalar Consumption Sync Activity:** Defines the business object for the activity to be created when the related measuring component is scalar. In the base package, this is set to "D1-RelMCREScalar".
- **Condition Bottom Range for Consumption Sync:** Defines the bottom of the condition range for measurements that will trigger consumption synchronization. In the base package, this is set to "300000" which is the bottom of the range that encompasses "estimate" conditions.
- **Condition Top Range for Consumption Sync:** Defines the top of the condition range for measurements that will trigger consumption synchronization. In the base package, this is set to "499999" which is the top of the range that encompasses "estimate" conditions.

Consumption Synchronization Activities and Algorithms

Consumption synchronization activities are created when the system determines that consumption synchronization is required for a related measuring component. These activities capture the following:

- **Consumption Reference Measuring Component:** The measuring component for which consumption synchronization will be performed.
- **Initiating Initial Measurement Data:** One or more initiating initial measurements that may trigger consumption synchronization.

- **Estimation Initial Measurement Data:** the successful estimation initial measurements created as a result of this activity (as well as the first estimation initial measurement that failed if applicable).

Consumption synchronization activities are created in the "Pending" state. The "Related MC Consumption Sync" batch process (D1-RMCRE) transitions the consumption synchronization activity from "Pending" to the "Re-Estimate" state.

The "Consumption Sync" Enter algorithm on the "Re-Estimate" state creates estimation initial measurements.

- For scalar measuring components, this initial measurement contains no measurement value.
- For interval measuring components, this initial measurement can contain a combination of "Regular" measurements and measurements that require estimation.

"Regular" measurements are derived from existing final measurement values that fall within the time span of the estimation initial measurement. For example, if the consumption synchronization process needs to re-estimate 3 intervals that were missing in the original initial measurement, the remaining intervals would be derived from the corresponding final measurement values for the measuring component.

Measurements that require estimation have a value of zero, and are based on (and correspond) to the measurements with condition codes that fell between the range defined for "consumption synchronization" when evaluated by the "Update Latest Measurement Date/Time" algorithms on the initial measurements (see **Initial Load and Manual Initial Measurement Algorithms** on page 6-28).

Example: An hourly interval initial measurement is received with 3 missing intervals (for the 10:00 AM, 11:00 AM, and 12:00 PM hours). These intervals are estimated and the entire initial measurement is finalized (the remaining 21 interval measurements are finalized with a condition of "regular"). A consumption synchronization estimation initial measurement created for this will contain three zero value intervals for the 10:00 AM, 11:00 AM, and 12:00 PM hours, while the remaining interval values will be copied from the "regular" final measurements.

The table below lists the algorithms for each of the consumption synchronization activities.

Activity Business Object	Algorithm
Interval Consumption Synchronization Activity Business Object: D1-RelMCREInterval	D1-RE-INTVMC (Interval MC Consumption Sync)
Scalar Consumption Synchronization Activity Business Object: D1-RelMCREScalar	D1-RE-SCMC (Scalar MC Consumption Sync)

These algorithms use the same set of parameters:

- **Condition Bottom Range for Consumption Sync:** Defines the bottom of the condition range for measurements that will trigger consumption synchronization. In the base package, this is set to "300000" which is the bottom of the range that encompasses "estimate" conditions.
- **Condition Top Range for Consumption Sync:** Defines the top of the condition range for measurements that will trigger consumption synchronization. In the base package, this is set to "499999" which is the top of the range that encompasses "estimate" conditions.
- **IMD Seeder Business Object:** Defines the business object to be used when creating estimation initial measurements. In the base package, this is set to "D1-IMDSeeder".

- **Condition For Missing Measurement:** Defines the condition that any measurement that should be synchronized (re-estimated/re-calculated). In the base package, this is set to "201000" which is the condition code for "missing".

Estimation Initial Measurement Algorithms

Once created by the consumption synchronization algorithms, the estimation initial measurements transition through lifecycle, moving from "Pending" to "VEE Ready" and (assuming no errors are encountered) to "VEE Complete".

The "Perform VEE for Estimation IMD" Enter algorithm the "VEE Complete" state of the estimation initial measurement business objects perform estimation calculations for measurements that require estimation, using VEE rules defined in the "Estimation VEE Group" for the measuring component.

In the base package, this is the D1-ESTM-VEE (Perform VEE for Estimation IMD) algorithm. This algorithm uses the following parameters:

- **VEE Group Role:** Defines the VEE group to execute. In the base package, this is set to "D1ES" (Estimation).
- **VEE Rule Category:** Defines the Rule Category, used to filter which rules are execute. In the base package, this is not defined.

VEE Rules

Several VEE rules have parameters that are used by the consumption synchronization process. This section outlines those rules and the specific parameters used by each.

See **Base Package VEE Validation Rules** (page 89) and **Base Package VEE Estimation Rules** (page 111) in the *Oracle Utilities Meter Data Management User's Guide* for more information about these VEE rules.

Interval Adjustment from Scalar

Interval Adjustment From Scalar rules adjust interval measurements based on existing scalar measurements such that the total of the intervals equals the scalar measurement value for the same time period.

This adjustment can be performed on either all intervals in the interval measurement, or only on those intervals that fall within a specified condition range. One or more final measurements must be present for the related measuring component between the start and end date/times of the current initial measurement.

In the context of consumption synchronization, this rule should be used with "secondary" interval measuring components that reference a scalar measuring component as the "Consumption Reference Measuring Component".

The table below lists the parameters used by this rule that impact the consumption synchronization process.

Parameter	Description
Bottom Range Condition Value	Specifies the bottom of the range of condition values for intervals that will be subject to adjustment.
Top Range Condition Value	Specifies the top of the range of condition values for intervals that will be subject to adjustment.

Parameter	Description
Interpolate Zero Quantity Intervals	A flag that indicates if any zero quantity intervals within the condition range should be estimated using linear interpolation prior to the application of the adjustment factor. Note: the set of intervals that will be interpolated is controlled by the parameters for the D2-INTADJSCA - 'Interval Adjustment Based on Related Scalar Measurement' algorithm.
Maximum Consecutive Hours to Interpolate	The maximum number of consecutive hours to be interpolated. If there are more hours to interpolate than the value of this parameter, an exception is generated based on the Interval Adjustment Exception parameter. Applicable only when Interpolate Zero Quantity Intervals is Yes.
Condition Value for Adjusted Intervals	The condition value assigned to intervals adjusted by this rule.
Adjust All Intervals for Empty IMD	A flag that indicates (yes or no) if the Intervals to Adjust parameter should be set to "All" when processing an initial measurement that contains all zero interval values with the conditions that fall into the condition range to adjust and the Intervals to Adjust parameter is set to "Restrict by Condition". This ensures that all zero intervals will be adjusted by this rule.
Required Related MC Measurements (%)	The percentage of time that must be covered by the measurement data of the related measuring component as compared to the time covered by the initial measurement that is being processed that must be present in order for the rule to execute. If a related measuring component does not have at least this amount of measurement data, the rule generates an exception based on the Insufficient Related MC Measurements Exception parameters. If there are more than one related measuring components, each of the related measuring components must have this minimum amount of measurement data. For example, if this parameter is set to 75% and the initial measurement being processed covers a period of 24 hours and the related measuring component's measurements only cover 12 of those hours, the system raises an exception (the related measuring component coverage is only 50%).

Parameter	Description
Override Condition for High Quality Source Measurements	<p>This is an optional set of parameters used to override condition codes for interval estimated measurements that are based on measurements from a related measuring component which are deemed to be of "high quality."</p> <ul style="list-style-type: none">• Minimum Condition Quality to Override: Specifies the minimum condition quality value for the related scalar measuring component's measurements. Measurements will be considered "high quality" if all applicable measurement conditions are equal to or greater than this value.• Override Condition Value: If all of the related scalar measurements are of "high quality" (based on the Minimum Condition Quality to Override parameter), this condition value will be used for the estimated measurement.
Insufficient Related MC Measurements Exception	<p>Defines the Exception Type and Severity for exceptions created when a related measuring component does not have enough measurement data to meet the value defined for the Required MC Measurement (%) parameter.</p>
Interpolization Failure Exception	<p>Defines the Exception Type and Severity for exceptions created when interpolization fails because the maximum number of hours to interpolate has been exceeded</p>

Scalar Calculation from Interval

Scalar Calculation From Interval rules calculate a scalar reading and consumption value from measurements for a related interval measuring component for the same time period as the initial measurement being estimated.

In the context of consumption synchronization, this rule should be used with "secondary" scalar measuring components that reference an interval measuring component as the "Consumption Reference Measuring Component".

The table below lists the parameters used by this rule that impact the consumption synchronization process.

Parameter	Description
Required Interval MC Measurement (%)	The percentage of time that must be covered by the measurement data of the related measuring component as compared to the time covered by the initial measurement that is being processed that must present in order for the rule to execute. If a related interval measuring component does not have at least this amount of measurement data, the rule generates an exception based on the Insufficient Related MC Measurements Exception parameters. If there are more than one related measuring components, each of the related measuring components must have this minimum amount of measurement data. For example, if this parameter is set to 75% and the initial measurement being processed covers a period of 24 hours and the related measuring component's measurements only cover 12 of those hours, the system raises an exception (the related measuring component coverage is only 50%).
Override Condition for High Quality Source Measurements	<p>This is an optional set of parameters used to override condition codes for scalar measurements that are based on measurements which are deemed to be of "high quality."</p> <ul style="list-style-type: none"> • Minimum Condition Quality to Override: Specifies the minimum condition quality value for the related interval measuring component's measurements. Measurements will be considered "high quality" if all applicable measurement conditions are equal to or greater than this value. • Override Condition Value: If all scalar measurements are of "high quality" (based on the Minimum Condition Quality to Override parameter), this condition value will be used for the estimated measurement.
Insufficient Related MC Measurements Exception	Defines the Exception Type and Severity for exceptions created when a related measuring component does not have enough measurement data to meet the value defined for the Required MC Measurement (%) parameter.

Sum Check

Sum Check rules compare the difference between the total consumption of an incoming initial measurement against the total consumption (based on final measurements) for a related measuring component on the same device over time.

When used with consumption synchronization, Sum Check VEE rules should be applied to the **secondary** measuring component.

If the difference exceeds a specified tolerance, the rule creates an exception. Related measuring components are those defined as “Consumption Reference Measuring Components” for the initial measurement’s measuring component.

When used with consumption synchronization, this rule can be used to flag measurements for consumption synchronization. To do this, set the Exception Type and Severity as follows:

- **Exception Type:** The exception type specified in the **Sum Check VEE Exception Type** parameter on the device configuration type.
- **Exception Severity:** Information

With these settings, exceptions will be logged, but the initial measurement can still be finalized, this triggering the consumption synchronization check (performed by the “Update Latest Measurement Date/Time” algorithm).

The table below lists the parameters used by this rule that impact the consumption synchronization process.

Parameter	Description
Required Related MC Measurements (%)	The percentage of time that must be covered by the measurement data of the related measuring component as compared to the time covered by the initial measurement that is being processed that must be present for a related measuring component in order for the rule to execute. If a related measuring component does not have at least this amount of measurement data, the rule generates an exception based on the Insufficient Related MC Measurements Exception parameters. If there are more than one related measuring components, each of the related measuring components must each have this minimum amount of measurement data. For example, if this parameter is set to 75% and the initial measurement being processed covers a period of 24 hours and the related measuring component’s measurements cover only 12 of those hours, the system raise an exception (the related measuring component coverage is only 50%).
Insufficient Related MC Measurements Exception	Defines the Exception Type and Severity for exceptions created when a related measuring component does not have enough measurement data to meet the value defined for the Required MC Measurement (%) parameter.

Scalar Proration

Scalar Proration rules estimate a scalar value by prorating the quantity of the measurement immediately following the initial measurement's "To Date/Time" (the subsequent measurement).

The table below lists the parameters used by this rule that impact the consumption synchronization process.

Parameter	Description
Minimum Condition to Preserve Measurement	If the initial measurement already has a measurement value and the condition for the measurement value is greater than or equal to this condition, the rule will exit and not perform proration calculations. If left blank, the rule will prorate existing measurements regardless of their condition code. This is to preserve estimations that may have been generated by other rules.
Minimum Related Measurement Condition	If there is a related interval measuring component and it has measurements for the time period covered by subsequent reading, the measurement must be at least this condition or greater to impact the proration.
IMD Created Condition Value	The condition value assigned to initial measurement data created by this rule.
Minimum Preceding Read Condition Quality	If specified, the prior reading must have a condition code greater than or equal to the specified condition code in order for the rule to execute. This is an optional parameter.
Minimum Subsequent Read Condition Quality	If specified, the subsequent reading must have a condition greater than or equal to the specified condition code in order for the rule to execute. This is an optional parameter.
Override Condition for High Quality Source Measurements	<p>This is an optional set of parameters used to override condition codes for scalar estimated measurements that are based on measurements which are deemed to be of "high quality."</p> <ul style="list-style-type: none"> Minimum Condition Quality to Override: Specifies the minimum condition quality value for the preceding and subsequent scalar measurements. Measurements will be considered "high quality" if both measurement conditions are equal to or greater than this value. Override Condition Value: If both the preceding and subsequent measurements are of "high quality" (based on the Minimum Condition Quality to Override parameter), this condition value will be used for the estimated measurement.
Reading Quality Exception	Defines the Exception Type and Severity for exceptions created if either the preceding or subsequent measurement does not meet the respective quality threshold specified.

Final Measurement Validation

Final Measurement Validation rules compare incoming initial measurement values with existing final measurement values for the same date/time range to determine which if any, are of a lesser quality (a lower condition code) and if final measurements should be created for them.

The table below lists the parameters used by this rule that impact the consumption synchronization process.

Parameter	Description
Measurement Replacement Validation	Defines the replacement condition code for measurements whose condition is less than the condition of existing final measurements
Replacement Condition Code	<p>When a measurement in the initial measurement is found to have a condition code that is of lesser quality than the existing measurement, it will be updated with the condition code specified here.</p> <p>Note: In order to ensure that final measurements are not overwritten by initial measurements with lesser condition codes than those of existing final measurements, this value should be set to the Condition Code for the 'System – No Read' parameter for the "normalize" Enter algorithm (based on the D1-NORM-IMD algorithm type) on the Finalized status of the initial measurement business object. The base package condition code of this type is "No Read - System (100000)."</p>
Measurement Replacement Exception	Defines the Exception Type and Severity for exceptions when an initial measurement's condition code is of lesser quality than the existing final measurement.

Configuration Example

This section provides a sample scenario and an example of how the system could be configured to meet the requirements of the scenario. Note that only the parameters that have a direct impact on the consumption synchronization process will be described.

Scenario Description

In this sample scenario, the system will be configured to have residential smart meters with one scalar initial measurement per day and one interval initial measurement per day. The scalar register is the “primary” channel and the related interval data will be adjusted to equal the scalar data if there is a difference between these two readings (actual or estimated values). Note: This configuration can be found in the Demonstration database for a customer called "Rea Estimate"

For any permutation of this relationship, the same basic building blocks will be used: a device with at least one interval measuring component and one register measuring component that are defined with a consumption reference relationship. The requirements of how meter data will be processed will dictate the intricacies of how each of these building blocks will be configured. This article will dive deep into one of the more common setups:

- One register measuring component read daily at midnight
- One interval measuring component with hourly intervals sent at midnight
- The register measuring component is the more reliable measuring component in the relationship

The sections below describe how the system could be configured to satisfy these requirements.

Device Configuration

The device configuration type for this scenario uses the following parameter values:

Parameter	Value
Keep Consumption Reference MC In Sync	Primary and Secondary
Minimum Condition to Re-estimate Primary MC	Regular
Sum Check VEE Exception Type	Consumption Differs from Reference Amount

The **Keep Consumption Reference MC In Sync** parameter defines which reference measuring components to synchronize.

- When "Secondary" is selected, consumption synchronization will be triggered for a secondary measuring component when new measurement data is received for the primary measuring component.
- When "Primary and Secondary" is selected, consumption synchronization can be triggered for the primary measuring component when new measurements are received for the secondary measuring component or re-estimation can be triggered for the secondary measuring component when new measurement data is received for the primary measuring component.

If the primary measurements are being updated from the secondary measuring component, a minimum condition code to trigger primary consumption synchronization from secondary is required (using the **Minimum Condition to Re-estimate Primary MC** parameter). This controls consumption synchronization of the primary measuring component based on measurements received for the secondary measuring component. For example, if scalar was primary and an interval initial measurement is received for a period where there were with existing scalar estimates and these were set to 501000 (Regular) then the interval measurement would need

a condition that was equal to or greater than 501000 to trigger a re-estimation of the scalar measurements.

The **Sum Check VEE Exception Type** field is populated with the exception type that will be generated by the primary measuring component's VEE rules for initial load and manual initial measurements. This exception will be used to test whether or not the consumption synchronization process should be initiated. If the secondary measuring component has measurement data for the time period and the consumption is within tolerance with the newly arrived primary measurement data, then there is no need to execute a consumption sync because the measuring components are already in agreement. Therefore, the consumption synchronization process will only be initiated when the initial measurement being processed for the primary measuring component encounters a "Sum Check VEE Exception Type".

Measuring Component Type

The scalar measuring component type (Auto-Read Register Type) must be setup to define when to mark estimates as "Do Not Use" for billing when actual scalar readings are received. The measuring component type can also be setup to allow consumption synchronization of measurements that are marked as "Do Not Use. "

The measuring component type for this scenario uses the following parameter values:

Parameter	Value
Ignore Estimates as IMD Start Reading	Yes
Flag Future Estimates as Do Not Use	No
Actuals or Corrections Initiate Re-Estimation	Yes

The **Ignore Estimates as IMD Start Reading** field determines if estimated measurements prior to the initial measurement being processed will be ignored and flagged as "Do Not Use".

The **Flag Future Estimates as Do Not Use** field determines if future estimates are marked as "Do Not Use" when new actual readings arrive.

The **Actuals or Corrections Initiate Re-Estimation** field determines whether any estimates that were set to "Do Not Use" will be re-estimated. When set to "Yes", any gaps created by the estimates will be re-filled by the periodic scalar estimation process.

Measuring Components

The system must be configured to define the primary and secondary measuring components. The primary measuring component is considered to be the more reliable measuring component on the meter and therefore the consumption it produces may be used to override the secondary measuring component. Furthermore, this measuring component's consumption will be used to validate the consumption of the secondary measuring component.

In this scenario the primary measuring component is the scalar register channel because it is seen as the more reliable source of consumption. The initial measurements for the primary measuring component should be processed prior to the initial measurements of the secondary measuring component so that its consumption will be available for the secondary measuring component's VEE rules.

The secondary measuring component is thought to have less reliable consumption data that may be overridden by consumption from the primary measuring component (when in conflict). Pending initial measurements for this measuring component are processed after the primary measuring component's initial measurements.

The measuring components for this scenario uses the following parameter values:

Primary Measuring Component

Parameter	Value
Information	ER-SM-018 / 1 / Electric kWh 1 Daily Read
Consumption Reference Measuring Component	

Secondary Measuring Component

Parameter	Value
Information	ER-SM-018 / 2 / Electric kWh 60min
Consumption Reference Measuring Component	ER-SM-018 / 1 / Electric kWh 1 Daily Read

Initial Measurement and Consumption Synchronization Algorithms

For this scenario, the algorithms on initial measurements and consumption synchronization activities will use the base package parameters (described previously).

VEE Groups and Rules

This section outlines VEE rules configurations that can be used with this consumption synchronization scenario.

Initial Load VEE - Interval Secondary Measuring Component

The following set of rules can be used to determine if synchronization on a secondary interval measuring component is needed.

- **Interval Interpolation:** This rule fills small gaps as needed
- **Interval Averaging Estimation:** This rule uses "like" days to fill larger gaps
- **Interval Adjustment From Scalar:** This rule adjusts the interval values to match the consumption from the scalar register measuring component.

Parameter	Value
Interval Adjustment From Scalar	<ul style="list-style-type: none"> • Intervals to Adjust: Restrict by Condition • Bottom Range Condition Value: System Estimate • Top Range Condition Value: Prorated Estimate • Interpolate Zero Quantity Intervals: No • Adjust All Intervals for Empty IMD: Yes • Required Scalar MC Measurement (%): 90
Override Condition for High Quality Source Measurements	<ul style="list-style-type: none"> • Minimum Condition Quality to Override: Prorated Estimate • Override Condition Value: Prorated Estimate

Parameter	Value
Insufficient Input Data Exception	<ul style="list-style-type: none"> Exception Type: Insufficient Data Input Exception Severity: Information
Interval Adjustment Exception	<ul style="list-style-type: none"> Exception Type: Interval Adjustment Exception Exception Severity: Information
Insufficient Related MC Measurements Exception	<ul style="list-style-type: none"> Exception Type: Insufficient Related MC Measurements Exception Severity: Information
Interpolization Failure Exception	<ul style="list-style-type: none"> Exception Type: Interval Size Discrepancy Exception Severity: Information
<ul style="list-style-type: none"> Sum Check: This rule ensures that the consumption from the two measuring components are in sync. 	
Parameter	Value
Sum Check	<ul style="list-style-type: none"> Percentage Tolerance: 1% Required Scalar MC Measurement (%): 90
Insufficient Input Data Exception	<ul style="list-style-type: none"> Exception Type: Insufficient Data Input Exception Severity: Information
Sum Check Exception	<ul style="list-style-type: none"> Exception Type: Consumption Differs from Reference Amount Exception Severity: Information
Insufficient Related MC Measurements Exception	<ul style="list-style-type: none"> Exception Type: Insufficient Related MC Measurements Exception Severity: Information
No Data Found Exception	<ul style="list-style-type: none"> Exception Type: No Historical Data Found Exception Severity: Information

Estimation VEE - Scalar Primary Measuring Component

The following set of rule can be used to perform consumption synchronization for a primary scalar measuring component. The order of the register estimation rules is important because it is desirable that the rule that will produce the highest quality estimate will be executed. The order in which these rules execute means that the estimation rules have the following order of precedence:

1. Scalar Calculation From Interval: This takes precedence only when the estimation it generates is of high quality (i.e. Prorated Estimate).
2. Scalar Proration: If Calculate Consumption From Interval did not generate an estimation or a Prorated Estimate, this rule will be responsible for the estimation.
3. Scalar Calculation From Interval: When this rule generates an estimate that is of lower quality than a Prorated Estimate, it will only be used if Scalar Proration did not generate an estimate.
4. Scalar Estimation: Will only execute when neither Calculate Consumption From Interval or Scalar Proration generated an estimate.

- **Scalar Calculation From Interval:** This rule creates or overrides existing measurements based on interval consumption. This ignores the current measurement value when estimating.

Parameter	Value
Scalar Calculation From Interval	<ul style="list-style-type: none"> • IMD Created Condition Value: System Estimate • Required Interval MC Measurement (%): 90
Override Condition for High Quality Source Measurements	<ul style="list-style-type: none"> • Minimum Condition Quality to Override: Regular • Override Condition Value: Prorated Estimate
Insufficient Input Data Exception	<ul style="list-style-type: none"> • Exception Type: Insufficient Data Input • Exception Severity: Information

- **Scalar Proration:** This rule prorates previous scalar estimates based on actual readings. This rule can be configured to skip processing if the estimates generated by a previous rule are of sufficient quality. This allows the Scalar Proration rule to override the Calculate Scalar from Interval rule when the incoming estimate is a "System Estimate" (301000) and Scalar Proration will generate a "Prorated Estimate".

Parameter	Value
Scalar Proration	<ul style="list-style-type: none"> • IMD Created Condition Value: System Estimate • Minimum Preceding Read Condition Quality: Prorated Estimate • Minimum Subsequent Read Condition Quality: Regular
Override Condition for High Quality Source Measurements	<ul style="list-style-type: none"> • Minimum Condition Quality to Override: • Override Condition Value: Prorated
Insufficient Input Data Exception	<ul style="list-style-type: none"> • Exception Type: Insufficient Data Input • Exception Severity: Information
Reading Quality Exception	<ul style="list-style-type: none"> • Exception Type: Reading Quality Exception • Exception Severity: Information

The **Scalar Proration** configuration indicates that proration will only occur if the measurement being estimated is between a measurement that is at least "Prorated Estimate" and a measurement that is at least "Regular".

No override information has been configured because the proration parameters have already limited this rule to only that data which would create a prorated estimate so this is unnecessary.

- **Scalar Estimation Rule:** This rule estimates register readings based on historical average daily usage values, but will only estimate if there is no estimated value yet present.
- **Final Measurement Validation:** This rule compares the incoming measurement from the initial measurement against the existing final measurement of the same measurement date/

time. If the initial measurement's measurement is of lesser quality than the existing final measurement, then it will trigger a VEE exception.

Parameter	Value
Measurement Replacement Validations	Replacement Condition Value: No Read - System
Measurement Replacement Exception	<ul style="list-style-type: none"> Exception Type: Measurement Replacement Exception Exception Severity: Information

The condition specified for the **Replacement Condition Value** parameter should be the same condition that is specified for the parameter "Condition Code for "System - No Read" on the "Normalize measurements" (D1-AUTO-NORM, D1-MNL-NORM and D1-SNORMIMD) algorithm. This algorithm is on the Finalized status for all initial measurement business objects.

Estimation VEE - Interval Secondary Measuring Component

The following set of rules can be used to perform synchronization on a secondary interval measuring component if needed.

- **Interval Interpolation:** This rule can be used to fill small gaps as needed
- **Interval Averaging Estimation:** This rule can be used to use "like" days to fill larger gaps
- **Interval Adjustment From Scalar:** This rule can be used to adjust all or portions of an interval curve to equal the scalar consumption values for the same period.

Parameter	Value
Interval Adjustment From Scalar	<ul style="list-style-type: none"> Intervals to Adjust: Restrict by Condition Bottom Range Condition Value: Missing Top Range Condition Value: System Estimate Interpolate Zero Quantity Intervals: No Maximum Hours to Interpolate: 13 Condition Value for Adjusted Intervals: System Estimate Adjust All Intervals for Empty IMD: Yes Required Scalar MC Measurement (%): 99
Override Condition for High Quality Source Measurements	<ul style="list-style-type: none"> Minimum Condition Quality to Override: Prorated Estimate Override Condition Value: Prorated Estimate
Insufficient Input Data Exception	<ul style="list-style-type: none"> Exception Type: Insufficient Data Input Exception Severity: Information
Interval Adjustment Exception	<ul style="list-style-type: none"> Exception Type: Interval Adjustment Exception Exception Severity: Information

Parameter	Value
Insufficient Related MC Measurements Exception	<ul style="list-style-type: none"> Exception Type: Insufficient Related MC Measurements Exception Severity: Information
Interpolization Failure Exception	<ul style="list-style-type: none"> Exception Type: Interpolization Failure Exception Exception Severity: Information

Note that the Interval Adjustment From Scalar parameters are different for this rule that for the rule used for the initial load rule.

- **Final Measurement Validation:** This rule can be used to compares the incoming measurement from the initial measurement against the existing final measurement of the same measurement date/time. If the initial measurement's measurement is of lesser quality than the existing final measurement, then it will trigger a VEE exception.

Parameter	Value
Measurement Replacement Validations	Replacement Condition Value: No Read - System
Measurement Replacement Exception	<ul style="list-style-type: none"> Exception Type: Measurement Replacement Exception Exception Severity: Information

The condition specified for the **Replacement Condition Value** parameter should be the same condition that is specified for the parameter "Condition Code for "System - No Read" on the "Normalize measurements" (D1-AUTO-NORM, D1-MNL-NORM and D1-SNORMIMD) algorithm. This algorithm is on the Finalized status for all initial measurement business objects.

Consumption Synchronization Examples

This section provides examples of consumption synchronization for some common scenarios, including:

- **Validating Across Interval Boundaries – Reading Times Vary**
- **Primary and Secondary Initial Measurements Processed Out of Order**
- **Standalone AMR Self Re-estimation**

Validating Across Interval Boundaries – Reading Times Vary

In many cases, register reading times do not occur on an interval boundary. The Sum Check and Interval Adjustment from Scalar VEE rules can be used to compare and adjust these measurements.

In this example, the Sum Check and Interval Adjustment from Scalar VEE rules are used to compare and adjust measurements from dependent measuring components that have different end times.

Step	Notes
A register read is received for the meter. However, the read was taken slightly late at 12:32AM instead of 12:00AM. This means that the consumption represented by this read will be slightly more than the related interval cut for the same day.	The register read was received outside of an interval boundary at 12:32AM When this quantity is compared to related interval data, the extra 32 minutes of consumption will need to be accounted for.
The interval cut for the same day is then received with two intervals missing. To estimate those missing intervals the register read is leveraged. Since the register read is for a larger time basis than the interval data, it is “intervalized” using the straight line method (i.e., the consumption is evenly distributed across all intervals). This allows the estimation process to account for the extra 32 minutes of consumption represented by the register read as compared to the interval cut.	Interval consumption is slightly less than the register consumption due to timing. The two missing intervals are estimated using the “intervalized” register consumption.

Step	Notes
Initial measurement trace logs can be examined to gain further insight into how the estimation was performed. the Interval Adjustment From Scalar and Sum Check rules have the ability to leverage register readings that do not align with the interval boundaries.	<p>The Interval Adjustment from Scalar rule is used to align the interval consumption for the estimated intervals to the register read.</p> <p>This scalar adjustment target is the result of “intervalizing” the register read using the straight line method and then taking only those intervals that aligned with our interval cut. Specifically, the consumption of 22.74 was divided evenly among 25 intervals with 0.9096 kWh each. Since the 25th interval does not match the interval initial measurement’s time basis, it was excluded. The result is 24 intervals at 0.9096 or 21.8304 kWh.</p> <p>The Sum Check rule is used to validate that the interval consumption and register consumption are within tolerance of each other.</p>

Primary and Secondary Initial Measurements Processed Out of Order

VEE rules are configured on the assumption of a certain processing order. This is to ensure that the data of the primary measuring component is available for the secondary measuring component to use for validation and estimation. When the secondary measuring component’s initial measurements are received first, this can reduce the efficacy of these VEE rules. The consumption synchronization process for measuring components is built to accommodate this common scenario and can generate an estimation initial measurement for the secondary measuring component as soon as more accurate data has been received for the primary measuring component

Step	Notes
A daily interval reading is received for 1/18, however, there is no register reads yet. The initial measurement is missing two intervals which will be estimated prior to finalization.	<p>The interval consumption is 26.03 kWh in total.</p> <p>Since there is no associated register read, missing intervals are estimated using interval interpolation.</p>
The IMD trace log can be used to see that the interpolation rule filled in the two missing intervals. An initial measurement exception was logged for the Sum Check rule since no associated register read exists to validate against. Note, the exception is only informational so this initial measurement still proceeded to finalization.	<p>An Interval Interpolation rule fills the gaps by estimating values for the missing intervals.</p> <p>Interval Averaging Estimation and Interval Adjustment from Scalar rules execute but do nothing.</p> <p>The Sum Check rule fails because there are no register reads for the same time period.</p>

Step	Notes
Subsequent to the interval initial measurement being finalized, the register read for the same day is received.	<p>This scalar consumption of 26.04 is 0.1 kWh less than the interval consumption already processed.</p> <p>Since the register channel is primary and this initial measurement is being processed after the interval data has already been finalized, the synchronization logic will check the interval measuring component for estimated measurements that fall within the time period covered by this register read. Given that there are two estimated intervals (as noted above), this initial measurement has spawned a Related MC Consumption Sync activity that will coordinate the re-estimation of those intervals.</p>
The Related MC Consumption Sync activity created by the register initial measurements remains in the "Pending" state until the Re-Estimation batch job is run or a user manually transitions it to re-estimate. Once transitioned, the logic will evaluate whether a new estimation initial measurement should be generated based on the measurements that exist for the Consumption Reference Measuring Component.s	<p>Due to the estimated intervals, the interval measuring component requires synchronization.</p> <p>Enabling tracing on the activity will expose the initial measurement trace logs in the event that the estimation initial measurement encounters issues.</p> <p>The Initiating Initial Measurement Data section contains any register initial measurements that contain measurements that may impact the interval measuring component's estimations. In this example there is only one initial measurement that had impact.</p>
The Related MC Consumption Sync batch process can be used to perform consumption synchronization.	The Related MC Consumption Sync batch process will transition all newly created activities from "Pending" to the "Re-Estimate" state. This will initiate the consumption synchronization process.

Step	Notes
<p>Once the batch is complete, the Related MC Consumption Sync activity will have been processed. Since no errors were encountered the activity is completed and a new estimation initial measurement will have been generated.</p>	<p>Each distinct time period to be re-estimated will be listed in the Estimation Initial Measurement Data section. The number of entries in this list is dependent on the number of initiating initial measurements and whether those initial measurements are contiguous or overlap. In this scenario, there was only one initiating initial measurement, therefore there is one entry. When the initial measurement creation is successful, the Success Flag will be populated (Yes). In the event of an error the relevant error information will be displayed for trouble shooting purposes.</p> <p>The activity's log tab will show that an estimation initial measurement was generated and provide a hyperlink to that initial measurement.</p>
<p>On the interval measuring component, there are now two initial measurements for the same day. One is the original initial load initial measurement and the other is the estimation initial measurement created by the consumption synchronization process.</p>	<p>Unlike a typical estimation initial measurement, there were pre-populated intervals with a non-zero quantity and a condition of "Regular."</p> <p>The two intervals that were missing in the original initial measurement have been estimated once more. This time with the benefit of the register read they have been adjusted so the total consumption for the interval initial measurement will match the total consumption for the register read. The 10AM interval has went from 1.049517 to 1.054862 kWh and the 11AM interval went from 1.020763 to 1.025962 kWh.</p> <p>The interval consumption of 26.04 kWh now matches register read of 26.04 kWh.</p>

Step	Notes
The IMD trace log shows that the intervals were once more populated by the Interval Interpolation rule. However, in this execution, the Interval Adjustment from Scalar rule is now able to retrieve scalar consumption and adjust the intervals in such a way that the total consumption for the day will match the register read.	Interpolation populates the initial estimations. Now that there is a register read to compare against, the Interval Adjustment from Scalar rule can calculate a multiplier that can be used to adjust the two estimated intervals.

Standalone AMR Self Re-estimation

The concept of re-estimating based on more accurate data is not exclusive to related measuring components. When a standalone AMR meter that has been configured for periodic estimation does not receive reads at the expected times, estimations will be generated. When these estimates are made, they generally must use predictive estimation methods (e.g., Scalar Estimation which uses like days). Once the next actual read is received for that measuring component, it makes sense that the estimate can be made more precise since there are now data points that exist before and after the reading.

The device setup for this scenario is slightly different from the "Rea Estimate " example in that there is only one channel, an automatically read register. This register is configured to receive multiple readings per day. The measuring component type for this scenario uses the following parameter values:

Parameter	Value
Ignore Estimates as IMD Start Reading	Yes
Flag Future Estimates as Do Not Use	No
Actuals or Corrections Initiate Re-Estimation	Yes
*Expected Hours Between Measurements	6
*First Daily Measurement Time	04:00 AM
*Early Measurement Threshold	01:00:00
*Late Measurement Threshold	01:00:00
Hours Before Estimation	72
Number of Hours to Estimate	24

These four parameters indicate that the register will be read four times: 4AM, 10AM, 4PM, and 10PM. Also, readings can be received up to an hour late or early and still satisfy these expected readings.

For this scenario, the meter has data up through 5/31/2013 11:00 PM

Step	Notes
On 6/4/2013 the periodic estimation process is executed to fill in any gaps with estimated measurements. As the measuring component Type is configured to wait 72 hours prior to estimating, only the missing measurements for 6/1/2013 will be gap filled.	<p>The Latest Read Date/Time now indicates that measurements are contiguous up through 6/1/2013 11PM.</p> <p>Four estimation initial measurements were generated for the 6/1/2013 4AM, 10AM, 4PM, and 10PM expected reads.</p> <p>Note: expected reads are expressed in standard time and but the user interface represents those times in "legal" (aka shifted for DST) time.</p>
The Measurements zone displays four estimated measurements for 11.53 kWh, 11.56 kWh, 11.58 kWh, and 11.61 kWh respectively. These estimates were derived using the Scalar Estimation rule which uses "like days" to generate an estimate (either from last year, or last reading).	
On 6/4/2013 after periodic estimation had completed, a new measurement was received from the head-end system. This measurement was for the 11AM expected read and was an actual read. As part of AMR self re-estimation, this resulted in all prior estimates that exist between the incoming actual and the last actual (on 5/31/2013 11PM) being set to "Do Not Use". Also, the Latest Read Date/Time field will be updated to reflect that there are now gaps in the measurement data that need to be estimated.	Having set the four estimated reads from 6/1 to "Do Not Use", the Latest Read Date/Time is rolled back to 5/13/2013 11PM to reflect the last time contiguous reads were present.
The Measurement zone displays the four 6/1/2013 measurements as "Do Not Use". The 6/2/2013 5AM measurement has a measurement quantity of 49 kWh which covers 5/31/2013 11PM through 6/2/2013 5AM.	
The gaps on 6/1/2013 will remain until the next time that periodic estimation is executed. For this example, it is run one more time on 6/4/2013 so only those same gaps were filled with estimates once more.	The Latest Read Date/Time is updated to 6/1/2013 11PM to reflect that the 6/1 gaps are now once more filled with estimates.

Step	Notes
<p>There were many initial measurements generated during this process. This is because for each estimate that was generated a manual (correction) initial measurement was required to correct the consumption that is calculated for the 6/2/2013 11AM reading.</p>	<p>The register channel was updated in the following sequence:</p> <ol style="list-style-type: none"> 1. A new estimation initial measurement was created for 6/01/2013 5AM 2. A manual initial measurement was generated for 6/02/2013 11AM to correct the consumption for that measurement. 3. A new estimation initial measurement was created for 6/01/2013 11AM 4. A manual initial measurement was generated for 6/02/2013 11AM to correct the consumption for that measurement. 5. A new estimation initial measurement was created for 6/01/2013 5PM 6. A manual initial measurement was generated for 6/02/2013 11AM to correct the consumption for that measurement. 7. A new estimation initial measurement was created for 6/01/2013 11PM 8. A manual initial measurement was generated for 6/02/2013 11AM to correct the consumption for that measurement.
<p>The four new estimates for 6/1/2013 were able to take advantage of the actual read that was received on 6/2/2013 11AM. The quantity for each measurement was calculated as a prorated amount of the 49 kWh of the 6/2/2013 reading. This results in each reading being 9.8 kWh with a condition of Prorated Estimate.</p>	

Initial Measurement Data In Detail

This section provides details concerning the initial measurement data objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom initial measurement data objects you create as part of your implementation. This section includes:

- A description of the D1-IMD maintenance object
- Lists of the base package initial measurement data business objects, including “lite” business objects
- Details concerning initial measurement data specific configuration options
- A sample initial measurement data business object (D1-InitialLoadIMDInterval)

Maintenance Object - D1-IMD

Initial measurement business objects use the D1-IMD maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-IMD
Description	Initial Measurement
Service Name	D1-INITMSRMTDATA (Initial Measurement Maintenance)
Tables	<ul style="list-style-type: none"> • D1_INT_MSRMT_DATA (Initial Measurement) - Primary • D1_INT_MSRMT_DATA_CHAR (Initial Measurement Characteristics) - Child • D1_INT_MSRMT_DATA_LOG (Initial Measurement Log) - Child • D1_INT_MSRMT_DATA_LOG_PARM (Initial Measurement Log Parameters) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Base Package Business Objects

The base package includes the following initial measurement data business objects:

Business Object Name	Description
D1-EstimationIMDInterval	Estimation IMD (Interval) Instances of this business object represent individual estimated interval initial measurements in the system.
D1-EstimationIMDScalar	Estimation IMD (Scalar) Instances of this business object represent individual estimated scalar initial measurements in the system.

Business Object Name	Description
D1-InitialLoadIMDInterval	Initial Load IMD (Interval) Instances of this business object represent individual interval initial measurements in the system.
D1-InitialLoadIMDScalar	Initial Load IMD (Scalar) Instances of this business object represent individual scalar initial measurements in the system.
D1-ManualIMDInterval	Manual IMD (Interval) Instances of this business object represent individual manually-created interval initial measurements in the system.
D1-ManualIMDScalar	Manual IMD (Scalar) Instances of this business object represent individual manually-created scalar initial measurements in the system.
D1-SyncIMDScalar	Sync Request IMD (Scalar) Instances of this business object represent individual scalar initial measurements in the system created via synchronization requests from an external system. Note: Initial measurements created using this business object are not subject to VEE processing. Because the data originate from an external application where it had already been loaded, this business object assumes the reading is valid.

The base package includes the following “lite” initial measurement data business objects:

Business Object Name	Description
D1-AuditList - IMD	Audit List Section Only (Lite)
D1-GenericIMDMain - IMD	IMD - Main Section Only LITE
D1-IMDLite	IMD LITE
D1-IMDParentLITE	IMD Parent LITE
D1-IMDPeriod	IMD Lite for High Quality Check
D1-IMDPostVEE	IMD Main and Post VEE LITE
D1-IMDPostVEERaw	IMD Main and Post VEE Raw LITE
D1-IMDPreAndPostVEE	IMD Pre and Post VEE - Lite
D1-IMDPreVEE	IMD Main and Pre VEE LITE
D1-IMDRawData	Initial Measurement Data Raw Lite
D1-IMDRetry	IMD Retry BO LITE

Business Object Name	Description
D1-IMDSeederLite	IMD Seeder Lite
D1-IMDTraceAndPostVEE	IMD Trace and Post VEE - Lite
D1-IMDTraceZone	IMD - Trace List Section Only LITE
D1-InitialLoadIMDIntervalRaw	Initial Load IMD (Interval) - Raw LITE
D1-InitialLoadIMDMain	Initial Load IMD - Main Section Only LITE
D1-PostVEE	Post VEE Only LITE
D1-PreVEE	Pre VEE Only LITE

The base package includes the following additional initial measurement data business objects:

Business Object Name	Description
D1-IMDSeeder	IMD Seeder (used to instantiate initial measurement business objects, based on the head-end system sending the measurement)
D1-SyncIMDScalar	Sync Request IMD (Scalar)

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by initial measurement data business objects.

Business Object Options

Initial measurement business objects can make use of the following business object options:

- **Apply Multiplier to Pre-VEE (Y/N):** This option indicates whether or not multiplier and other consumption factors are to be applied to Pre-VEE measurement data.
- **Channel Status Code Mapping:** This option defines an extendable lookup used for mapping vendor-specific channel status codes to standard codes.
- **Initial Measurement Data Type:** This option defines the data type for initial measurement data created from this business object. Valid values are based on the D1_IMD_TYPE_FLG lookup, and include the following:

Field Value	Description
D1IL	Initial Load
D1MO	Manual
D1ES	Estimation

- **Interval Status Code to Condition Mapping:** This option defines the Interval Status Code extendable lookup that can be used to map status codes for initial measurement data created from this business object.
- **Override Automated Retry (Y/N):** This option indicate whether or not the execution of the IMD Automated Retry process should be forced regardless of the value of the IMD Automated Retry flag in the BO schema.

Example Initial Measurement - D1-InitialLoadIMDInterval

The table below lists the details of the D1-InitialLoadIMDInterval initial measurement data business object.

Option/Field	Description
Business Object	D1-InitialLoadIMDInterval
Description	Initial Load IMD (Interval)
Maintenance Object	D1-IMD (Initial Measurement Data)
Application Service	D1-INITLOADIMDBOAS (Initial Load IMD Interval BO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> • Apply Multiplier to Pre-VEE (Y/N): Y • Initial Measurement Data Type: D1IL (Initial Load) • Override Automated Retry (Y/N): Y • Display UI Map: D1-IMDDisplay (Initial Measurement Data - Display) • Portal Navigation Option: d1imdsaTabMenu (Initial Measurement) • Display Map Service Script: D1-IMDDisp (Initial Measurement Data Display) • Maintenance UI Map: D1-IMDMaint (Initial Load IMD (Interval) - Maintenance)
Algorithms	<ul style="list-style-type: none"> • Post-Processing: D1-AUD-QTYUE (Audit IMD Quantity Changes made to Quantity and Set User-Edited Flag) • Validation: D1-IMD-VAL (Validates Status Condition for Delete) • Validation: D1-IMD-COMM (Validate Initial Measurement Data Common Input) • Validation: D1-INT-SPEC (Validate Interval Initial Measurement Data Input)
Lifecycle	<ul style="list-style-type: none"> • Pending (Initial) • Additional Mapping (Interim) • Mapping Error (Interim) • VEE Ready (Interim) • Error (Interim) • Removed from Processing (Final) • VEE Complete (Interim) • Exception (Interim) • Discarded (Final) • Force Complete (Interim) • Finalized (Final)

Use the Business Object portal to view additional details concerning this business object.

Meter Data Management Base Package Initial Measurement Data Business Objects

The Oracle Utilities Meter Data Management base package includes the following initial measurement data business objects:

Business Object Name	Description
D2-IMDHoursLateLite	IMD Hours Late Lite

Measurements In Detail

This section provides details concerning the measurement objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom measurement objects you create as part of your implementation. This section includes:

- A description of the D1-MSRMT maintenance object
- Lists of the base package measurement business objects, including “lite” business objects
- Details concerning measurement-specific configuration options
- A sample measurement business object (D1-Measurement)

Maintenance Object - D1-MSRMT

Measurement business objects use the D1-MSRMT maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-MSRMT
Description	Measurement
Service Name	D1-MEASUREMENT (Measurement Maintenance)
Tables	<ul style="list-style-type: none">• D1-MSRMT (Measurement) - Primary• D1-MSRMT_CHAR (Measurement Characteristics - Child)

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Base Package Device Business Objects

The base package includes the following measurement business objects:

Business Object Name	Description
D1-Measurement	Measurement Instances of this business object represent individual final measurements stored in the system.

The base package includes the following “lite” device business objects:

Business Object Name	Description
D1-MeasurementParentLITE	Measurement Parent LITE
D1-MSRMTLite	Measurement LITE

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by measurement business objects.

Business Object Options

Measurement business objects can make use of the following business object options:

- **Measurement Log Business Object:** This option defines the business object that will be used to log changes that have occurred to the measurement.

Example Device - D1-Measurement

The table below lists the details of the D1-Measurement device business object.

Option/Field	Description
Business Object	D1-Measurement
Description	Measurement
Maintenance Object	D1-MSRMT (Measurement)
Application Service	D1-MEASUREMENT (Measurement MO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> • Measurement Log Business Object: D1-MeasurementLog (Measurement Log) • Display UI Map: D1-MeasurementDisplay (Measurement - Display) • Portal Navigation Option: d1meassaTabMenu (Measurement) • Display Map Service Script: D2-MsrmtDisp (Measurement Data Display)
Algorithms	<ul style="list-style-type: none"> • Audit: D1-AMSRMTLOG (Formats the standard description of a Measurement) • Information: D1-MSRMTINFO (Measurement Information)
Lifecycle	<ul style="list-style-type: none"> • OK (Initial) • Re-Derive Values (Interim)

Use the Business Object portal to view additional details concerning this business object.

Meter Data Management Base Package Measurement Business Objects

The Oracle Utilities Meter Data Management base package includes the following aggregated measurement business objects:

Business Object Name	Description
D2-MeasuredQuantityMsrmt	Measured Quantity Measurement Instances of this business object represent individual measured quantity aggregated measurements stored in the system.

Business Object Name	Description
D2-QualityCountMsrmnt	Quality Count Measurement Instances of this business object represent individual quality count aggregated measurements stored in the system.
D2-TimelinessCountMsrmnt	Timeliness Count Measurement Instances of this business object represent individual timeliness count aggregated measurements stored in the system.
D2-TimelinessQualityMsrmnt	Timeliness Quality Measurement Instances of this business object represent individual timeliness quality aggregated measurements stored in the system.

Device Type Service Quantities in Detail

This section provides details concerning the device quantity service quantity objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom device quantity service quantity objects you create as part of your implementation. This section includes:

- A description of the D1-DVTPSQ maintenance object
- Lists of the base package device quantity service quantity business objects, including “lite” business objects
- Details concerning device quantity service quantity-specific configuration options
- A sample device quantity service quantity business object (D1-Transformer)

Maintenance Object - D1-DVTPSQ

Service point business objects use the D1-DVTPSQ maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-DVTPSQ
Description	Device Type Service Quantity
Service Name	D1-DVTPSQ (Device Type Service Quantity Maintenance)
Tables	<ul style="list-style-type: none"> • D1_DVC_TYPE_SQ (Device Type Service Quantity) - Primary • D1_DVC_TYPE_SQ_CHAR (Device Type SQ Characteristic) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Device Type Service Quantity Business Objects

The Service and Measurement Data Foundation base package includes the following device type service quantity business objects:

Business Object Name	Description
D1-AvgDailyEstItemConsumption	<p>Average Daily Estimated Item Consumption</p> <p>Instances of this business object represent individual device type service quantities defined in the system.</p>

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by device type service quantity business objects.

System Events

Device type service quantity business objects can make use of the following system events:

- Calculate Device Type Service Quantity** This system event defines the algorithm used to calculate service quantities for device types when calculating usage for service points at which devices that make use of device type service quantities are installed (such as items). The Algorithm Entity for available algorithms is “Device Type SQ (BO) - Calculate Device Type Service Quantity.”

Example Device Type Service Quantity - D1-AvgDailyEstItemConsumption

The table below lists the details of the D1-AvgDailyEstItemConsumption device type service quantity business object.

Option	Description
Business Object	D1-AvgDailyEstItemConsumption
Description	Average Daily Estimated Item Consumption
Maintenance Object	D1-DVTPSQ (Device Type Service Quantity)
Application Service	D1-DVTPSQ (Device Type Service Quantity MO)
Instance Control	Allow New Instances
Algorithms	<ul style="list-style-type: none">Calculate Device Type Service Quantity: D2-CALC-EDA (Calculate Consumption from Estimated Daily Average)Information: D1-QUANINFO (Average Estimated Daily Service Quantity Information)

Use the Business Object portal to view additional details concerning this business object.

Configuring Custom Measurement Data

This section provides high-level overviews of the steps involved in configuring custom objects to define initial measurements, reader remark types, final measurements, and device type service quantities. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

Note: The procedures below focus on specific configuration tasks and options related to each of the objects described in this chapter, and do not address all the steps involved in creating business objects, UI maps, algorithms, etc. For more information about these subjects, refer to the Oracle Utilities Application Framework documentation.

Configuring Custom Initial Measurements

Configuring custom initial measurement objects involves the following steps:

1. Design the initial measurement business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom initial measurement-related configuration objects required for your business objects, including:

Business Object Options: Create business objects for the following business object options:

- IMD Status Extendable Lookup
3. Create your initial measurement business objects, referencing the configuration objects created above as appropriate.

Configuring Reader Remark Types

Configuring reader remark types involves the following:

1. Determine the specific types of reader remarks that meter readers will be able to specify when reading meters.

Note that reader remark types do not allow readers to enter free text. They must select from a pre-defined list of remark types. Examples might include “BROKEN_SEAL”, “DAMAGED_METER”, “EVIDENCE_OF_TAMPERING”, “DOG_ON_PREMISES”, etc.

2. Define a reader remark type (device event type) for each type of reader remark identified in step 1 above. Define the following for each:

- Reporting Category
- Eligible for Processing
- To Do Types
- To Do Roles
- Service Issue Monitor Type

Refer to the Oracle Utilities Service and Measurement Data Foundation User’s Guide for more information about defining reader remark types.

Configuring Custom Measurements

Configuring custom measurement objects involves the following steps:

1. Design the measurement business objects you will need to create for your implementation, including the data and processing required for each.

2. Create the custom measurement-related configuration objects required for your business objects, including:
Business Object Options: Create business objects for the following business object options:
 - Measurement Log Business Object
3. Create your initial measurement business objects, referencing the configuration objects created above as appropriate.

Configuring Custom Device Type Service Quantities

Configuring custom device type service quantity objects involves the following steps:

1. Design the device type service quantity business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom device type service quantity-related configuration objects required for your business objects, including:
System Events: Create algorithms for the following system events:
 - Calculate Device Type Service Quantity

Create your device type service quantity business objects, referencing the configuration objects created above as appropriate.

Chapter 7

Validation, Editing, and Estimation

This chapter provides descriptions of validation, editing, and estimation groups and rules, including:

- **Understanding Validation, Editing, and Estimation**
- **VEE Groups In Detail**
- **VEE Rules In Detail**
- **VEE Eligibility Criteria In Detail**
- **VEE Exceptions In Detail**
- **Configuring VEE Groups, Rules, Eligibility Criteria, and Exceptions**

Understanding Validation, Editing, and Estimation

This section describes the validation, editing, and estimation (or VEE) process used by the Service and Measurement Data Foundation and related products, including Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway. This section includes:

- **Overview of the Validation, Editing, and Estimation Process**
- **VEE Rules and VEE Groups**
- **VEE Exceptions**

Overview of the Validation, Editing, and Estimation Process

As noted in **Chapter 6: Measurement Data**, once received into the system, initial measurements are subject to validation, editing, and estimation. This process involves the following:

- **Validation:** Validates that the initial measurement data is within expected tolerances, and is correct
- **Editing:** If the initial measurement data is wrong in some way, the data can be automatically changed.
- **Estimation:** If initial measurement data is incomplete (for example, if one or more interval values within an interval measurement are missing), missing values can be automatically estimated.

The values recorded in an initial measurement can change during this process, and exceptions can be raised if initial measurements are wrong in some way (such as out-of-tolerance quantities, incorrect values, etc.).

Beyond raising an exception, the VEE process can also transition an initial measurement to an Exception state if it detects a problem that it is not able or allowed to correct.

VEE Rules and VEE Groups

The specific validation, editing, and estimation processing performed on initial measurement data is defined in individual VEE rules, each performing a specific set of validation logic. Examples of VEE rules include:

- **Interval Size Validation:** Checks to ensure that the interval size of the incoming initial measurement data matches the value defined in the measuring component type
- **Multiplier Check:** Checks to ensure that the device multiplier value of the incoming initial measurement data matches the multiplier value stored on the measuring component
- **Unit of Measure Check:** Checks to ensure that the Unit of Measure (UOM) of the incoming initial measurement data matches the UOM specified on the measuring component's type

The base package contains many VEE rules you can use in your implementation, and you can also create your own custom VEE rules.

Some VEE rules create exceptions if the initial measurement data doesn't fall within parameters specified by the rule. Other rules override measurements, changing measurement values as dictated by the rule's parameters. Some rules can both create exceptions and override the measurement as part of a single process. By convention, VEE rules change the Post VEE quantities of initial measurement data, but VEE rules can change ANYTHING on an initial measurement.

VEE Groups

VEE groups are collections of VEE rules that are applied to initial measurement data. During the VEE process, the system executes the VEE rules defined in each VEE group. The rules within a

VEE groups are defined in a specific sequence, allowing control over the order in which the rules are executed.

VEE groups can be associated to a specific measuring component, or to a measuring component type (or both). VEE groups associated with a measuring component type are applied to all measuring components of that type, while those associated to a specific measuring component are applied only to that measuring component. VEE groups associated to a measuring component override those assigned to a measuring component type.

Effective Dates

Every VEE rule has an effective period. Rules will only be applied if the initial measurement's start date is within the rule's effective period. For example, an Interval Spike Check rule with a Start Date of 11/15/2010 will only be applied if the start date of the initial measurement is on or after 11/15/2010.

This allows you to update the specifics of a rule without removing the previous version of the rule. For example, you might change the tolerance of an Interval Spike Check rule from 1.2 to 1.5 as of a certain date. However, for initial measurement data for the period prior to the change, you would want to use the tolerance for the original version of the rule (1.2) instead of the new tolerance (1.5).

Eligibility Criteria

Each VEE rule may optionally have eligibility criteria that controls if the rule is applied. This feature can greatly reduce the number of VEE groups you need to create, because it allows a single VEE group to have conditional VEE rules based on eligibility criteria (rather than requiring a distinct VEE group for every combination of VEE rules).

For example, you might create a rule that compares interval consumption against related scalar consumption (such as might be the case with a device with both interval and scalar measuring components). This rule might use eligibility criteria that specifies that the rule is only applied if the initial measurement's measuring component has a corresponding scalar register measuring component.

Another example might be a rule that compares the current consumption against standard consumption for measuring components of a certain type for the first six months after installation. You might create eligibility criteria that specifies that the rule is only applied if the measuring component's device configuration has been installed at a service point for less than six months.

Generic Utility Rules - Tools For Creating VEE Groups

While most VEE rules are used to validate the usage (or some other attribute) in an initial measurement, Oracle Utilities Service and Measurement Data Foundation also provides a number of "generic utility" VEE rules that can be used when configuring VEE groups. These include:

- Execute VEE Group Rules
- VEE Group Matrix (Factor) Rules
- Exception Handler Rules
- Successful Termination Rules

Execute VEE Group Rules - Reusing Groups Of Rules

A common situation in many implementations is in which several rules are to be applied to multiple different types of measuring components. For example, you might want to perform device identifier validations, multiplier checks, and UOM checks on all measuring components.

One way to meet this requirement would be to repeat these three rules in multiple VEE groups. However, this solution becomes hard to maintain if changes to the rules are required (or if new "global rules" are introduced) as each group would have to be updated.

Instead of this, you can create a VEE rule that executes the rules in a referenced VEE group. Rules of this type are called Execute VEE Group rules. Rules of this type can have effective dates and eligibility criteria, just like all VEE rules.

Using the example above, you could create a group called “Rules for All MCs” that contains a device identifier validations rule, a multiplier check rule, and a UOM check rule, and then reference the “Rules for All MCs” group in a Execute VEE Group rule.

Execute VEE Group rules can be “nested.” That is, a group executed by a Execute VEE Group rule can, in turn, execute the rules in another group, and so on.

VEE Group Matrix (Factor) Rules - Using Factors To Implement Dynamic VEE Groups

Another situation likely to occur in many implementations is where specific rules may need to be applied to measurement data based on specific criteria, such as geography. For example, some geographic territories may have unique VEE rules in addition to rules that are applied to all geographic territories.

This requirement could be implemented using eligibility criteria, such as only applying a rule (or a group of rules) if the service point for an initial measurement is located in a hot summer area. If there are limited number of these unique rules, this solution is suitable, but if there are many territories and each territory has several unique rules, an implementation of this sort would become hard to maintain (and slow to execute) as the VEE group would have many rules with varying eligibility criteria.

Instead of this, you can create a VEE rule that dynamically executes another group's rules based on specific conditions. For example, the VEE process can be configured to execute different VEE rules based on where the service point is located, or the number of tamper events in the last 6 months, or the type of customer, or the meter's head-end system, etc. Rules of this type are called VEE Group Matrix (Factor) rules. These rule and can have effective dates and eligibility criteria, just like all VEE rules.

Factors are used to implement the dynamic selection of VEE group (note the term factor is intentionally generic as factors can be used for other purposes). Factors used for these rules have a Factor Class of “VEE group,” and use some unique rules:

- VEE group factors reference a characteristic type (with pre-defined values).
- VEE group factors reference an algorithm that retrieves or derives the value of the characteristic type at runtime.
- Factor values for a VEE group factor are effective-dated pairings of a characteristic value and a corresponding VEE group.

At run time, the rule retrieves / derives the characteristic value for the factor's characteristic type and then finds the VEE group associated with the respective characteristic value.

Factors can be related to any real or dynamic attribute, so rules of this type are very flexible. For example:

- **Real Attribute:** you could create a rule that executes a VEE group based on the head-end system of the device.
- **Dynamic Attribute:** you could create a rule that executes a VEE group based on the number of tamper events linked to the measuring component in the last 180 days, executing one group if there are 6-10 events (a characteristic value of 6-10), and another if there are more than 10 events (a characteristic value of 10+). The number of tamper events is dynamically calculated at execution time and is compared to the characteristic values defined for the factor, and executes the appropriate VEE group. In this example, if the count of tamper events was anything less than six, no VEE group would be executed.

Exception Handler Rules

Exception Handler rules are described in the section below on VEE exceptions.

Successful Termination Rules

Successful Termination rules are described in the section below on VEE exceptions.

VEE Roles

Initial measurement data can come from different sources, such as a head-end system or estimation processes, or it can be manually created by a user (to override or estimate consumption). Measurement data from these different sources might use different VEE rules. For example:

- Initial measurements sent a head-end system might use strict VEE rules
- Initial measurements created by a user (to override or estimate consumption) may use less strict rules
- Initial measurements created by the system to estimate consumption have very few (if any) VEE rules

Applying different numbers and types of VEE rules based on the source of the initial measurement data could be implemented using eligibility criteria (e.g., only apply a rule if the initial measurement data's source is X) or factor-based rules (e.g., the factor's characteristic is the initial measurement data's source), but both of these techniques are potentially difficult to maintain if there are many source-dependent rules.

Instead of that approach, you can define different VEE groups for different source, or roles. The three base package roles are:

- **Estimation:** Used for initial measurement data estimated by the system
- **Initial Load:** Used for initial measurement data received from a head-end system or import process
- **Manual Override:** Used for initial measurement data manually created by a user

A measuring component's Measuring Component Type can define "fallback" VEE groups for each of these roles. In addition, an individual measuring component can specify a VEE group for each role. If the measuring component doesn't have a VEE group specified for a role, the "fallback" VEE group defined for the measuring component type is used.

VEE Exceptions

Each VEE rule defines an exception type and severity that specify how exceptions are tracked by the system. When an initial measurement fails a validation, an exception of the type specified for the failed VEE rule is created. A single initial measurement can have multiple exceptions, one (or more) for each rule the measurement fails. This allows users to see all of the problems detected during the VEE process.

Exception Types

Each exception has an exception type. Exception types allow you to distinguish between different exceptions based on the rule that triggered them. Exception types can be created each VEE rule, at a more general level exception types, such as "Insufficient Data" to be used to signify that a measurement didn't have sufficient data for the VEE rule to execute.

Exception Categories

There are three categories, or severities of exceptions:

- **Info:** Used to highlight something interesting, but not sufficient to cause the initial measurement to be put into the Exception state. Exceptions of this category can be used to report on the frequency of interesting, but not fatal issues.
- **Issue:** Used to report a problem that will prevent the initial measurement from being finalized. Multiple "issue exceptions" can be created during VEE processing. If at least one issue exists after all rules have been applied, the initial measurement is transitioned to the Exception state.
- **Terminate:** Used to report a severe issue that will cause the VEE process to stop and the initial measurement to be transitioned immediately to the Exception state. Only one terminate exception can be issued (as the first one causes VEE to stop on an initial measurement).

Exceptions and To Do Entries

In addition to exceptions, VEE processing can also trigger the creation of To Do Entries related to failed validations.

If Issue or terminate exceptions exist for an initial measurement, a To Do Entry is created when the initial measurement is transitioned to the Exception state. The To Do Type and default To Do Role of this To Do Entry are defined on the Enter system event for the Exception state of the business object used to define the initial measurement.

To Do Entries created in this way can be routed to different roles depending on the exception's message category and number (using the To Do Type's Message Overrides tab).

Exception Handler Rules - Aborting When There Are Too Many Issues

There may be times when an implementation's requirements are to terminate processing for any initial measurement that contains a pre-defined number of exceptions. Exception Handler VEE rules can issue a "terminate exception" if they detect too many exceptions. This is useful when individual exceptions are not sufficient to stop VEE processing.

The criteria used by this rule can simply reference a number of exceptions of a given exception type, or can specify more complex AND/OR criteria that must be satisfied before VEE processing is terminated. For example, processing might terminate when 3 exceptions of one type AND 2 exceptions of another type have been issued, or if 2 exceptions of one type OR 2 exceptions of a different type have been issued.

The terminate exception created by Exception Handler rules can be of a specific exception type. In addition, Exception Handler rules can also create a different type of To Do Type and To Do Role than the default.

Exception Handler rules can be placed at any point throughout a VEE group where each rule can reference different exception types.

Successful Termination Rules

There may be times when an implementation's requirements are to successfully terminate processing for any initial measurement that passes a pre-defined set of validations before accumulating a pre-defined number of exceptions. For example, a set of validation rules can be executed early in the overall sequence of rules that proves that the data is good enough to use, such that no further rules need to be executed. In this case, implementations might want to terminate the VEE process to save on execution time rather than execute further rules that won't ultimately affect the data. This is accomplished through Successful Termination rules.

The criteria used by Successful Termination rules can simply reference a number of exceptions of a given exception type, or can specify more complex AND/OR criteria that must be satisfied before VEE processing is terminated. For example, processing might terminate when less than 3 exceptions of one type AND less than 2 exceptions of another type have been issued, or if less than 2 exceptions of one type OR less than 2 exceptions of a different type have been issued.

Successful termination rules can be placed at any point throughout a VEE group where each rule can reference different exception types.

Available Actions for Initial Measurements with Exceptions

Users have a number of options for dealing with initial measurements with exceptions.

- After correcting the cause of the issues that triggered the exceptions, a user can re-VEE the initial measurement.
- A user can discard the initial measurement.
- A user can edit the Post VEE quantities (if necessary) and manually complete the initial measurement. This will cause final measurements to be created using the contents of the Post VEE quantities.

Note: No VEE processing is performed on manually completed initial measurement data.

Regardless of the action taken by the user, the system will complete any open To Do Entries that created when the initial measurement entered the Exception state.

Exceptions Are Not Deleted

Note that exceptions are not deleted when an initial measurement is adjusted or corrected. After any issues are corrected or the initial measurement is overridden (or manually completed), the exceptions persist (in the Closed state) for reporting purposes.

VEE Groups In Detail

This section provides details concerning the VEE group objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom VEE group objects you create as part of your implementation. This section includes:

- A description of the D1-VEEGROUP maintenance object
- Lists of the base package VEE group business objects, including “lite” business objects
- A sample VEE group business object (D1-VEEGroup)

Maintenance Object - D1-VEEGROUP

Device business objects use the D1-VEEGROUP maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-VEEGROUP
Description	VEE Group
Service Name	D1-VEEGROUP (VEE Group Maintenance)
Tables	<ul style="list-style-type: none">• D1_VEE_GRP (VEE Group) - Primary• D1_VEE_GRP_CHAR (VEE Group Characteristics - Child• D1_VEE_GRP_L (VEE Group Language) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package VEE Group Business Objects

The Service and Measurement Data Foundation base package includes the following VEE group business objects:

Business Object Name	Description
D1-VEEGroup	VEE Group

The Service and Measurement Data Foundation base package includes the following additional VEE group business objects:

Business Object Name	Description
D1-VEEGroupBundlingAddBO	Bundling Add BO for VEE Group
D1-VEEGroupPhysicalBO	Physical BO for VEE Group

Example VEE Group - D1-VEEGroup

The table below lists the details of the D1-VEEGroup device business object.

Option/Field	Description
Business Object	D1-VEEGroup
Description	VEE Group
Maintenance Object	D1-VEEGROUP (VEE Group)
Application Service	D1-VEEGROUP (VEE Group MO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none">• Display UI Map: D1-VEEGroupDisplay (VEE Group - Display)• Portal Navigation Option: d1veegrpTabMenu (VEE Group)• Maintenance UI Map: D1-VEEGroupMaint (VEE Group - Maintenance)

Use the Business Object portal to view additional details concerning this business object.

VEE Rules In Detail

This section provides details concerning the VEE rule objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom VEE rule objects you create as part of your implementation. This section includes:

- A description of the D1-VEERULE maintenance object
- Lists of the base package VEE rule business objects, including “lite” business objects
- Details concerning VEE rule-specific configuration options
- A sample VEE rule business object (D2-IntervalSpikeCheck)
- A list of base package VEE rules, including the algorithm / algorithm type and a brief description of each

Maintenance Object - D1-VEERULE

Device business objects use the D1-VEERULE maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-VEERULE
Description	VEE Rule
Service Name	D1-VEERULE (VEE Rule Maintenance)
Tables	<ul style="list-style-type: none"> • D1_VEE_RULE (VEE Rule) - Primary • D1_VEE_RULE_CHAR (VEE Rule Characteristics) - Child • D1_VEE_RULE_L (VEE Rule Language) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package VEE Rule Business Objects

The Service and Measurement Data Foundation base package includes the following VEE rule business objects:

Business Object Name	Description
D1-CommonVEERule	Common VEE Rule (VEE rule business object used as a parent business object for other VEE rule business objects)
D1-DuplicateIMDCheck	Duplicate IMD Check Instances of this business object represent individual Duplicate IMD Check rules defined in the system.
D1-GenericVEERule	Generic VEE Rule (generic VEE rule business object used as a parent business object for all other VEE rule business objects)

Business Object Name	Description
D1-VEERuleExceptionHandler	VEE Rule Exception Handler Instances of this business object represent individual exception handler VEE rules defined in the system.
D1-VEERuleGroupFactor	VEE Group Matrix (Factor) Instances of this business object represent individual VEE Group Matrix (Factor) VEE rules defined in the system.
D1-VEERuleReferredVEEGroup	Execute VEE Group Instances of this business object represent individual Execute VEE Group VEE rules defined in the system.
D1-VEERuleSuccessTermination	Successful Termination Instances of this business object represent individual Successful Termination VEE rules defined in the system.

The Service and Measurement Data Foundation base package includes the following “lite” VEE rule business objects:

Business Object Name	Description
D1-VEERuleGroupFactorLite	VEE Rule Group Factor LITE
D1-VEERuleParentLITE	VEE Rule Parent LITE

The Service and Measurement Data Foundation base package includes the following additional VEE rule business objects:

Business Object Name	Description
D1-VEERuleBundlingAddBO	Bundling Add BO for VEE Rule
D1-VEERuleExecReSequence	VEE Rule Execution Resequencing (used when resequencing VEE rules in a VEE group)
D1-VEERulePhysicalBO	Physical BO for VEE Rule

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by VEE rule business objects.

System Events

VEE rule business objects can make use of the following system events:

- **Apply VEE Rule:** This system event defines the algorithm to use when executing the VEE rule.

Other Options

VEE rules use various parameters and properties. These options are specified when creating VEE rules based on a VEE rule business object, and include the following:

Exception Types

Exception types define the properties common to exceptions. When creating VEE rules, you might create an exception type for each rule. You might also create more general exception types, such as "Insufficient Data" to be used to signify that a measurement didn't have sufficient data for the VEE rule to execute.

Utility VEE Rules

Oracle Utilities Service and Measurement Data Foundation includes "utility" base package VEE rule types that can be used when configuring VEE groups and rules. This section outlines the configuration options you need to configure before you can create rules of these types.

Execute VEE Group: Execute VEE Group rules reference a VEE group. You must create the VEE group to reference before can create rules of this type.

Exception Handler: Exception Handler rules are used to define options and logic to terminate the VEE process when a set of user configured criteria are met. VEE rules of this type can be included in a group to specify how exceptions are handled for that group, and allow for creation of a single "parent" exception for the group. Exception Handler rules use the following options:

- **To Do Type:** An override To Do Type for To Do Entries created as a result of the rule. This To Do Type is used instead of the default (specified in the Enter algorithm on the Exception state of the initial measurement data business object's lifecycle).
- **To Do Role:** An override To Do Role for To Do Entries created as a result of the rule. This To Do Role is used instead of the default (specified in the Enter algorithm on the Exception state of the initial measurement data business object's lifecycle).
- **Exception Type:** The Exception Type for exceptions created by this rule.

VEE Group Matrix (Factor): VEE Group Factor rules are used to define business logic to allow reference to a factor (of type VEE group) where the values of the factor are a list of VEE groups. This allows creating a VEE rule that can select from a list of VEE groups (referred to as a matrix) whose rules to execute next. VEE Group Matrix (Factor) rules use the following options:

- **Factor:** The factor referenced by the rule. The factor must have a Factor Class of VEE Group (i.e. it must be based on the VEE group factor business object).
- **Characteristic Type:** The characteristic type referenced by the factor. This characteristic type must be one with pre-defined values.
- **Characteristic Type Values:** Specific values for the characteristic type. These are the values retrieved and evaluated to determine the VEE group whose rules should be executed. These must be values that can be retrieved from some object (device, service point, etc.) related to the measuring component whose initial measurement data is being validated by this rule.
- **Characteristic Source Algorithm:** The algorithm used to retrieve the characteristic value (which in turn determines the VEE group whose rules should be executed). The base package includes the following algorithm types that can be used when creating this algorithm:
 - Factor Characteristic Source - Device (D1-FCSDEVICE)
 - Factor Characteristic Source Measuring Component (D1-FCSMC)
 - Factor Characteristic Source - Service Point (D1-FCSSP)
 - Factor Characteristic Source - Usage Subscription (D1-FCSUS)
- **VEE Groups:** The VEE groups associated with each characteristic value.

Successful Termination: Successful termination rules are used when an implementation's requirements are to successfully terminate processing for any initial measurement that passes a pre-defined set of validations before accumulating a pre-defined number of exceptions. Successful termination rules use the following options:

- **Exception Type:** The Exception Type(s) to check for when the rule is executed.

Example VEE Rule - D2-IntervalSpikeCheck

The table below lists the details of the D2-IntervalSpikeCheck VEE rule business object.

Option/Field	Description
Business Object	D2-IntervalSpikeCheck
Description	Interval Spike Check
Maintenance Object	D1-VEERULE (VEE Rule)
Parent Business Object	D1-GenericVEERule (Generic VEE Rule)
Lifecycle Business Object	Generic VEE Rule
Application Service	D1-VEERULE (VEE Rule MO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> Display UI Map: D2-IntervalSpikeCheckDisplay (Interval Spike Check - Display) Portal Navigation Option: d1veerleTabMenu (VEE Rule) Maintenance UI Map: D2-IntervalSpikeCheckMaint (Interval Spike Check - Maintenance)
Algorithms	<ul style="list-style-type: none"> Apply VEE Rule: D2-INTSPKCHK (Interval Spike Check) Validation: D2-INTSPKVLD (Interval Spike Check Minimum Number of Intervals Validation)

Use the Business Object portal to view additional details concerning this business object.

Meter Data Management Base Package VEE Rule Business Objects

The Oracle Utilities Meter Data Management base package includes the following VEE rule business objects:

Business Object Name	Description
D2-EnsureIMDExistsforSibling	Ensure IMD Exists for Sibling MCs
D2-FinalMeasurementValidation	Final Measurement Validation
D2-IntervalAdjustmentFrmScalar	Interval Adjustment from Scalar
D2-IntervalAveragingEstimation	Interval Averaging Estimation
D2-IntervalInterpolationEst	Interval Interpolation Est
D2-IntervalProfileEstimation	Interval Profile Estimation
D2-IntervalReplacementRule	Interval Replacement Rule
D2-IntervalSizeValidation	Interval Size Validation
D2-IntervalSpikeCheck	Interval Spike Check
D2-NegativeConsumptionCheck	Negative Consumption Check
D2-RaiseMissingQuantityExcp	Raise Missing Quantity Exception
D2-RegisterMultiplierCheck	Multiplier Check
D2-ScalarCalcFromInterval	Scalar Calculation From Interval
D2-ScalarEstimation	Scalar Estimation
D2-ScalarProfileEstimation	Scalar Profile Estimation
D2-ScalarProration	Scalar Proration
D2-ScalarReplacementRule	Scalar Replacement Rule
D2-SumCheck	Sum Check
D2-UOMCheck	Unit of Measure Check
D2-VEERuleHighLowCheck	High/Low Check
D2-ZeroConsumptionCheck	Zero Consumption Check

Base Package VEE Rules Summary

The following table lists the back package VEE rules. Each of these VEE rules is provided as a business object and corresponding algorithm/algorithm type.

VEE Rule	Business Object	Algorithm / Algorithm Type	Description
Common VEE Rule	D1-CommonVEERule	N/A	Parent VEE rule for other VEE rules, contains basic VEE rule schema elements
Duplicate IMD Check	D1-DuplicateIDMCheck	D1-DUPIMDCHK	Checks whether the current IMD is a duplicate of one already received for the same measuring component.

VEE Rule	Business Object	Algorithm / Algorithm Type	Description
Generic VEE Rule	D1-GenericVEERule	N/A	Parent VEE rule for all other VEE rules, contains basic VEE rule schema elements
VEE Rule Exception Handler	D1-VEERuleExceptionHandler	D1-AVEER-EEH	Allows a user to define a set of exception count criteria that, if met, results in the termination of the VEE process.
VEE Group Matrix (Factor)	D1-VEERuleGroupFactor	D1-AVEER-FCT	Uses the factor configured on the rule to determine a VEE group whose rules are to be executed
Execute VEE Group	D1-VEERuleReferredVEEGroup	D1-AVEER-RFG	Executes the rules for the VEE group configured on the rule
Device Multiplier Check	D2-RegisterMultiplierCheck	D2-REGMULCHK	Checks to ensure that the device multiplier value of the incoming initial measurement data matches the multiplier value stored on the measuring component.
Energy Sum Check	D2-SumCheck	D2-SUM-CHK	Checks the difference between the total consumption of incoming initial measurement data against the total consumption (based on final measurements) for one or more related measuring components on the same device over the same time period.
Ensure IMD Exists for Sibling MCs	D2-EnsureIMDExistsforSibling	D2-ENSIMDMC	Validates that initial measurements exist for all other measuring components associated to the same device configuration as the current measuring component, for the same period of time as the current initial measurement
Final Measurement Validation	D2FinalMeasurementValidation	D2-VLMSRCOND	Compares incoming initial measurement values with existing final measurement values for the same date/time range to determine which if any, are of a lesser quality (a lower condition code) and if final measurements should be created for them.
High/Low Check	D2-VEERuleHighLowCheck	D2-HILO-CHK	Compares consumption of incoming initial measurement data against historical data as a means of assuring the reasonableness of the data.
Interval Averaging Estimation	D2-IntervalAveragingEstimation	D2-INTAVGEST	Estimates missing interval initial measurement data based on averaging historical data for the same measuring component
Interval Interpolation Est	D2-IntervalInterpolationEst	D2-INTINTEST	Estimates missing interval initial measurement data using linear interpolation
Interval Profile Estimation	D2-IntervalProfileEstimation	D2-INTPROEST	Estimates missing interval initial measurement data based on a specified profile (a profile factor)

VEE Rule	Business Object	Algorithm / Algorithm Type	Description
Interval Replacement Rule	D2-IntervalReplacementRule	D2-INTREPL	Determines whether or not an incoming initial measurement will replace any existing measurements and if so, whether or not to allow this to happen.
Interval Size Validation	D2-IntervalSizeValidation	D2-INTSIZVAL	Checks to ensure that the interval size of the incoming initial measurement data matches the value defined in the measuring component type.
Interval Spike Check	D2-IntervalSpikeCheck	D2-INTSPKCHK	Examines incoming initial measurement data to identify intervals with conspicuously high usage relative to surrounding intervals.
Negative Consumption	D2-NegativeConsumptionCheck	D2-NCON-CHK	Checks for negative consumption.
Raise Missing Quantity Exception	D2-RaiseMissingQuantityExcp	D2-RAIMISQTY	Raises an exception if a percentage of the measurements within an initial measurement are marked with a condition code that falls within a specified range
Scalar Estimation	D2-ScalarEstimation	D2-SCALAREST	Estimates missing scalar initial measurement data based on historical data for the same measuring component.
Scalar Proration	D2-ScalarProration	D2-SCLRPRORT	Estimate a scalar value by prorating the quantity of the measurement immediately following the initial measurement's "To Date/Time" (the subsequent measurement).
Scalar Replacement Rule	D2-ScalarReplacementRule	D2-SCAREPRL	Identifies whether or not a scalar reading will completely replace an existing measurement and if so, whether or not to allow this to happen.
Unit Of Measure Check	D2-UOMCheck	D2-UOMCHK	Checks to ensure that the Unit of Measure (UOM) of the incoming initial measurement data matches the UOM specified on the measuring component's type.
Zero Consumption Check	D2-ZeroConsumptionCheck	D2-ZEROCNCHK	Checks for zero consumption.

Use the Business Object portal and Algorithm portal (or Application Viewer) to view additional details about these VEE rules. The algorithm types listed above use a set of base package measurement services. See **Appendix A: Measurement Services** for a list of available base package measurement services.

VEE Eligibility Criteria In Detail

This section provides details concerning the VEE eligibility criteria objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom VEE eligibility criteria objects you create as part of your implementation. This section includes:

- A description of the D1-VEEELIGCR maintenance object
- Lists of the base package VEE eligibility criteria business objects, including “lite” business objects
- Details concerning VEE eligibility criteria-specific configuration options
- A sample VEE eligibility criteria business object (D1-VEEEligibilityCriteria)

Maintenance Object - D1-VEEELIGCR

VEE eligibility criteria business objects use the D1-VEEELIGCR maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-VEEELIGCR
Description	VEE Eligibility Criteria
Service Name	D1-VEEELIGCR (VEE Eligibility Criteria)
Tables	<ul style="list-style-type: none"> • D1_VEE_ELIG_CRIT (VEE Eligibility Criteria) - Primary • D1_VEE_ELIG_CRIT_CHAR (VEE Eligibility Criteria Characteristics) - Child • D1_VEE_ELIG_CRIT_L (VEE Eligibility Criteria Language) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package VEE Eligibility Criteria Business Objects

The Service and Measurement Data Foundation base package includes the following VEE eligibility criteria business objects:

Business Object Name	Description
D1-VEEEligibilityCriteria	VEE Eligibility Criteria Instances of this business object represent individual eligibility criteria defined in the system.

The Service and Measurement Data Foundation base package includes the following additional VEE eligibility criteria business objects:

Business Object Name	Description
D1-VEEEligCritBundlingAddBO	Bundling Add BO for VEE Eligibility Criteria
D1-VEEEligCritPhysicalBO	Physical BO for VEE Eligibility Criteria

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by VEE eligibility criteria business objects.

System Events

VEE eligibility criteria business objects can make use of the following system events:

- **Apply VEE Rule Eligibility Criteria:** This system event defines the algorithm to use to apply eligibility criteria to a VEE rule.

Example VEE Eligibility Criteria - D1-VEEEligibilityCriteria

The table below lists the details of the D1-SmartMeter device business object.

Option/Field	Description
Business Object	D1-VEEEligibilityCriteria
Description	VEE Eligibility Criteria
Maintenance Object	D1-VEEELIGCR (VEE Eligibility Criteria)
Application Service	D1-VEEELIGCR(VEE Eligibility Criteria MO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none">• Portal Navigation Option: d1veerleTabMenu (VEE Rule)• Maintenance UI Map: D1-VEEEligibilityCritMaint (VEE Eligibility Criteria - Maintenance)
Algorithms	<ul style="list-style-type: none">• Apply VEE Rule Eligibility Criteria: D1-ECF-APECT (Evaluate Criteria Field - Apply Eligibility Criteria)

Use the Business Object portal to view additional details concerning this business object.

VEE Exceptions In Detail

This section provides details concerning the VEE exception objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom VEE exception objects you create as part of your implementation.

This section includes:

- A description of the D1-VEEEXCP maintenance object
- Lists of the base package VEE exception business objects, including “lite” business objects
- A sample VEE exception business object (D1-VEEException)

Maintenance Object - D1-VEEEXCP

VEE exceptions business objects use the D1-VEEEXCP maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-VEEEXCP
Description	VEE Exception
Service Name	D1-VEEEXCP (VEE Exception Maintenance)
Tables	<ul style="list-style-type: none"> • D1_VEE_EXCP (VEE Exception) - Primary • D1_VEE_EXCP_CHAR (VEE Exception Characteristics) - Child • D1_VEE_EXCP_PARM (VEE Exception Parameter) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package VEE Exception Business Objects

The Service and Measurement Data Foundation base package includes the following VEE exception business objects:

Business Object Name	Description
D1-VEEException	VEE Exception Instance of this business object represent exceptions created as a result of VEE processing of initial measurements.

Example VEE Exception - D1-VEEException

The table below lists the details of the D1-SmartMeter device business object.

Option/Field	Description
Business Object	D1-VEEException
Description	VEE Exception

Option/Field	Description
Maintenance Object	D1-D1-VEEEXCP (VEE Exception)
Application Service	D1-D1-VEEEXCP (VEE Exception MO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none">• Display UI Map: D1-VEEExceptionDisplay (VEE Exception - Display)• Display Map Service Script: D1-VEEExcpn (Display VEE Exception Message Details)
Algorithms	<ul style="list-style-type: none">• Information: D1-VEXCPINFO (VEE Exception Information)

Use the Business Object portal to view additional details concerning this business object.

Meter Data Management Base Package VEE Exception Business Objects

The Oracle Utilities Meter Data Management base package includes the following VEE exception business objects:

Business Object Name	Description
D2-VEEExceptionServiceMonitor	VEE Exception - Monitor Service Point Instances of this business object represent exceptions generated as a result of VEE processing, whose exception types have been configured to trigger the creation of a service issue monitor.

Configuring VEE Groups, Rules, Eligibility Criteria, and Exceptions

This section provides high-level overviews of the steps involved in configuring custom VEE groups and rules. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

This section also provides information related to creating instances of the “generic utility” VEE rules described earlier in this chapter.

Note: The procedures below focus on specific configuration tasks and options related to each of the objects described in this chapter, and do not address all the steps involved in creating business objects, UI maps, algorithms, etc. For more information about these subjects, refer to the Oracle Utilities Application Framework documentation.

Configuring Custom VEE Groups

Configuring custom VEE groups involves the following steps:

1. Design the VEE group business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom VEE group-related configuration objects required for your business objects.
3. Create your VEE group business objects, referencing the configuration objects created above as appropriate.

Configuring Custom VEE Rules

Configuring custom VEE rules involves the following steps:

1. Design the VEE rule business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom VEE rule-related configuration objects required for your business objects, including:

System Events: Create algorithms for the following system events:

- Apply VEE Rule

Options: Create data as appropriate for the following options used when creating VEE rules:

- Exception Types
- Generic Utility VEE Rules: Define the following options used with “generic utility” VEE rules
 - Referred VEE Group: VEE groups to be referenced by these rules.
 - Exception Handler: To Do Type, To Do Role, and Exception Type used by these rules.
 - VEE Group Matrix (Factor): Factor, Characteristic Type and Values, Characteristic Source Algorithm, VEE groups

3. Create your VEE rule business objects, referencing the configuration objects created above as appropriate.

Note: VEE rule business objects should reference D1-GenericVEERule as their Parent Business Object.

Configuring Custom VEE Eligibility Criteria

Configuring custom VEE eligibility criteria involves the following steps:

1. Design the VEE eligibility criteria business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom VEE eligibility criteria-related configuration objects required for your business objects, including:

System Events: Create algorithms for the following system events:

- Apply VEE Rule Eligibility Criteria
3. Create your VEE eligibility criteria business objects, referencing the configuration objects created above as appropriate.

Configuring Custom VEE Exceptions

Configuring custom VEE exceptions involves the following steps:

1. Design the VEE exception business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom VEE exception-related configuration objects required for your business objects.
3. Create your VEE exception business objects, referencing the configuration objects created above as appropriate.

Creating Generic Utility VEE Rules

This section outlines the steps involved in creating instances of the “generic utility” VEE rules. Refer to the Oracle Utilities Service and Measurement Data Foundation online help and user’s guide for general procedures used in creating VEE rules.

Creating Execute VEE Group VEE Rules

Use the following procedure to create Execute VEE Group VEE rules:

1. Create the VEE group to which the rule will belong.
2. Create the VEE group that the rule will reference
3. Create the VEE rule referencing the group created in the previous step

Creating VEE Group Matrix (Factor) VEE Rules

Use the following procedure to create VEE Group Matrix (Factor) VEE rules:

1. Create the Characteristic Type and Values to be used by the factor that will be referenced by the rule.
2. Create the Characteristic Source Algorithm to be used by the factor that will be referenced by the rule.
3. Create the VEE Groups to be associated to the characteristic values.
4. Create the Factor that will be referenced by the rule.
5. Create the Factor Values for the factor, each referencing an effective-dated characteristic value/VEE group pairings.
6. Create the rule, referencing the factor

Creating Exception Handler VEE Rules

Use the following procedure to create Exception Handler VEE rules:

1. Create the To Do Type to be used by the rule.
2. Create the To Do Role to be used by the rule.
3. Create the Exception Type to be used by the rule.
4. Create the rule, including:
 - To Do Type, To Do Role, and Exception Type created in the previous steps.
 - Criteria comparison for the rule.

Creating Successful Termination VEE Rules

Use the following procedure to create Successful Termination VEE rules:

1. Create the Exception Type to be used by the rule.
2. Create the rule, including:
 - Exception Type created in the previous steps.
 - Criteria comparison for the rule.

Chapter 8

Device Communication and Device Events

This chapter provides descriptions of entities related to device communications, including activities, communications, and completion events. This chapter also describes entities related to device events. This chapter includes:

- **Understanding Activities**
- **Understanding Device Events**
- **Understanding Service Tasks**
- **Activities in Detail**
- **Device Events in Detail**
- **Service Tasks in Detail**
- **Configuring Activities and Device Event Objects**
- **Configuring Service Tasks**

Understanding Activities

This section provides an overview of activities and how they are used in the Service and Measurement Data Foundation and related products, including Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway.

About Activities

Activities are records of a communication or event related to a device, measuring component, service point or other entity in the system. Examples of activities include meter read downloads (for manually read meters) or “last gasp” messages sent by devices when they detects they are about to power down.

Activity Type Categories

Activities and activity types fall into several categories based on how they are used in the system. The table below lists the base package activity type categories and how they are used.

Activity Type Category	Activities of this type category are used to:
Bulk Activity	Initiate a smart meter command for a group of meters using Oracle Utilities Smart Grid Gateway
Command Request	Initiate a smart meter command for an individual meter using Oracle Utilities Smart Grid Gateway
Consumption Sync	Trigger related measuring component estimation synchronization
Device Event Activity	Represent a device event. These are used when viewing usage and events with the Oracle Utilities Meter Data Management 360 Degree View.
Dimension Scanner	Trigger dimension scanning when aggregating usage, statistics, and other data.
Extract Request	Retrieve usage and device events from a head-end system using Oracle Utilities Smart Grid Gateway
Field Activity*	Initiate work to be performed in the field, typically sent to a field work system (such as Oracle Utilities Mobile Workforce Management).
Meter Read Download Activity	Download a list of meters to be read
Multi Device Command Request	Initiate a smart meter command for multiple meters (for head-end systems that support multi-device commands) using Oracle Utilities Smart Grid Gateway
Non-Dispatchable Activity*	Record an activity that need not be routed to an external system. Example: Wait for Measurement. Activities of this type are typically created by a request orchestration activity.
Orchestration Maintenance*	Cancel or update an existing request orchestration activity.
Payload Statistics	Calculate processing statistics for usage and device event import processing with Oracle Utilities Smart Grid Gateway
Request Orchestration*	Record a request for service, such as enabling/disabling service at a service point.

Activity Type Category	Activities of this type category are used to:
Suppression	Suppress sending of device events to subscribing systems (such as Oracle Utilities Network Management System)
Usage Transaction Correction Processor	Trigger correction processing for usage transactions when corrected usage is received that would impact an existing usage transaction

*Activities of these types are used by Oracle Utilities Service Order Management. Refer to the *Oracle Utilities Smart Grid Gateway Service Order Management Configuration Guide* for more information.

Understanding Device Events

This section provides an overview of device events and how they are used in the Service and Measurement Data Foundation and related products, including Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway.

Device Events

Device events are events of some sort that have taken place relative to a device, and can include power outages, power restorations, tampering alerts, command completions, and other events.

Attributes used to define device events include the following:

- **Device Event Date/Time:** the date and time of the event. For events with a duration, such as a power outage, this is the start date and time of the duration.
- **Device Event End Date/Time:** the end date and time of events with durations (such as power outages). Not applicable to events with no duration, such as a tampering alert or power restoration.

In addition, device events also reference details specific to the head-end system that sent the event, including the following:

- **Sender:** the head-end system (defined as a service provider in SGG) from which the event was sent.
- **External Sender ID:** the external ID for the head-end system that sent the event.
- **External Event Name:** the external, head-end-specific name for the event. This name is translated into a “standard” event name within SGG.
- **External Source Identifier:** an identifier for the source of the event.

Standard Event Names

When the system is loading a device event, it first determines which head-end system sent it, and then determines which device produced it. The system then maps the device event name sent by the head-end system to a standard event name that corresponds to the event. Standard event names are beneficial because the event names sent by head-end systems vary, even when the events are essentially the same. What one head-end system calls “Primary Power Down,” another system may call “Outage at Endpoint.” Creating a standard event name, such as “Power Outage - Device,” makes it easier to send the device events to an external system such as Oracle Utilities Meter Data Management. In this way, the external systems that receive the events do not need to be configured to accept all of the possible names for a single type of event.

Standard event names are configured on the Standard Event Name extendable lookup. A standard event name should be configured for each unique business purpose served by the device events that the system receives from the associated head-end systems.

Device Event Mapping

Each head-end-specific event name (which is also called an external event name) that the system will receive from a given head-end should be mapped to a standard event name. This mapping is configured using a device event mapping extendable lookup. Each head-end system should have its own extendable lookup to define event name mapping in order to prevent possible conflicts between mappings. For example, head-end systems A and B might both use the same event name, such as the code “1”, but this event might need to be mapped to “outage” for head-end system A but “tamper” for head-end system B.

The device event mapping extendable lookup business object is configurable. Each Oracle Utilities Smart Grid Gateway adapter includes a device event mapping lookup business object for the supported head-end system.

Device Event — Additional Processing

Certain device event business objects can be used to update master data as a result of device events received into the system. For example, when a utility's field worker arms a meter, the resulting device event can trigger an update to the device's arming status in the Oracle Utilities application. This processing is initiated by the Execute - Additional Processing system event algorithm during the Additional Processing status of the business object's lifecycle.

If a standard device event requires additional processing, the algorithms that execute the processing should be specified in the Additional Processing Algorithms list on the Standard Event Name extendable lookup for the event. The base package includes the additional processing algorithm Arm Meter (D1-ARM-METER).

The following base package business objects are configured to execute additional processing:

Business Object Name	Description
D1-DeviceEvtComResp	Device Event Communication Response Instances of this business object represent communications created in response to device events
D1-PairedEventFirstDeviceEvent	Device Event - Paired Event (First) Instances of this business object represent individual paired event (first) device events in the system.
D1-PairedEventLastDeviceEvent	Device Event - Paired Event (Last) Instances of this business object represent individual paired event (last) device events in the system.
D1-StandardDeviceEvent	Standard Device Event Instances of this business object represent individual standard device events in the system.

Standard vs. Paired Device Events

Some device events represent events that occur without a duration, such as a tampering alert, while other device events represent events with a duration, such as an outage (the duration being the time between the start and end of the outage period).

Device events with no duration are defined using “standard” device event business objects. The Service and Measurement Data Foundation base package provides a sample standard device event business object that can be extended as needed to support implementation-specific requirements. Instances of the standard event business object represent individual device events received into the system.

Device events with a duration are defined using “paired event” business objects, with the first of the pair representing the start of the event, and the last of the pair representing the end of the event. The Service and Measurement Data Foundation base package provides a sample set of paired device event business objects that can be extended as needed to support implementation-specific requirements. Events of this type can be configured to create or complete activities that represent the event. For example, an outage event might create an outage activity that is completed when power is restored. In this example, the outage event would be the first of the pair, while the power restoration event would be the last of the pair.

When pairs of events arrive in rapid succession (such as a last gasp followed quickly by a power restoration), these “paired event” business objects are designed to prevent them from being sent to subscribing applications.

Sending Device Events to Subscribing Systems

When device events are received, they are typically passed onto to another subscribing system, such as Oracle Utilities Meter Data Management, a customer information system (such as Oracle Utilities Customer Care and Billing), an outage system (such as Oracle Utilities Network Management System), or some other application.

The means of sending device event information to subscribing system is defined in the “How to Send Device Event Related Information” processing method for the service provider representing the subscribing system. Device event information can be sent via outbound communication business object, outbound communication, or batch process.

The method for sending device events to subscribing systems (business object, outbound message, or batch process) can be defined for each device event category, and can be overridden for individual device event types (including the ability to exclude specific device event types within a category). In addition, a default event processing method can be also be configured that applies when methods of transmission aren't specified at the individual category level.

The “Subscribe to Device Event” business service is used to process device event subscription requests, and allow external applications to manage the categories of events they receive.

Creating Service Issue Monitors and Service Investigative Orders

Device events can be configured to create service issue monitors and service investigative orders. The service issue monitor type to create is defined on the device event type. When a device event is received, an Enter algorithm on the device event's Processed state creates the service issue monitor, based on the Service Issue Monitor Type specified for the device event type.

Understanding Device Event Processing

This section provides an overview of the process that takes place when device events are received. For each step in the process, the table below provides a brief description of the processing that takes place, and lists the specific objects used by the Service and Measurement Data Foundation

Note that the process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process.

Step	Process	Service and Measurement Data Foundation Objects
1.	<p>The head-end system sends a payload of device events to middleware components. The payload contains both standard and paired device events.</p> <p>Oracle Utilities Smart Grid Gateway adapters use Oracle Service Bus. (OSB) as the middleware components for import of usage readings and device events,</p>	
2.	The middleware parses the payload into individual device events, and then transforms each from the format used by the head-end system into the format used by SMDF and SGG.	
3.	The middleware then invokes the Device Event Seeder XAI Inbound Service for each device event. The Device Event Seeder XAI Inbound Service is mapped to the Device Event Seeder business object.	XAI Inbound Service: D1-DeviceEventSeeder (D103879193)

Step	Process	Service and Measurement Data Foundation Objects
4.	The Device Event Seeder business object executes a number of pre-processing algorithms to derive the service provider and device ID, and shift event date/times to Standard time.	<p>Pre-Processing Algorithm: D1-SPRID (Service Provider Identification)</p> <p>Pre-Processing Algorithm: D1-DEVICEID (Device Identification)</p> <p>Pre-Processing Algorithm: D1-SHEVTDITM (Shift Event Date/Times to Standard)</p>
5.	<p>The Device Event Seeder business object determines the type of device event business object to create based on the Device Event Mapping business object (extendable lookup) defined for the head-end system service provider.</p> <p>Different head-end systems may use different names for the same type of event. This process maps device event names sent by the head-end system to standard device event names.</p>	<p>Pre-Processing Algorithm: D1-DETBOLD (Device Event Type and Business Object Identification)</p> <p>Processing Method: D1-HowToProcessDeviceInfo (How to Process Device Related Information)</p> <p>Device Event Mapping BO: D1-DeviceEventMappingLookup</p>
6.	<p>An instance of the appropriate device event business object is created in the system for each device event received.</p> <p>For paired events, the first event business objects may create activities to represent the event.</p>	<p>Standard Device Event BO: D1-StandardDeviceEvent</p> <p>Paired (First) Device Event BO: D1-PairedEventFirstDeviceEvent (Device Event - Paired Event (First))</p> <p>Paired (Last) Device Event BO: D1-PairedEventLastDeviceEvent (Device Event - Paired Event (Last))</p>
7.	If Additional Processing algorithms have been defined for the device event's standard name (in the Standard Event Name extendable lookup), these algorithms are executed when the device event enters the Additional Processing state.	Standard Event Name BO: D1-StdEventNameLookup
8.	<p>If there are systems that subscribe to device events (defined in a processing method for the system service provider), the device event is sent to those systems.</p> <p>Sending device event information is triggered by an Enter algorithm on the "Processed" Status of the device event business object's lifecycle.</p> <p>Note: Sending device event information may involve configuration of XAI senders, outbound message types, and external systems. Refer to the Oracle Utilities Application Framework for more information about using XAI.</p>	<p>Processing Method: D1-HowToProcDveEvtsInformation (How to Process Device Event Related Info)</p>

Time Zone Translation and Conversion

If your organization receives device events from a source that provides its data in a different time zone than the one in which the data will be stored in Oracle Utilities Meter Data Management, a translation can be performed to translate the time zone from an external time zone identifier to one configured within the application. This translation is defined via the Head-End Time Zone to Standard Mapping extendable lookup.

The "Shift Event Date/Times to Standard" pre-processing algorithm on the Device Event Seeder business object (D1-SHEVTDTTM) ensures that the date/times supplied with device event seeder data are in standard time and are converted to the correct time zone.

The algorithm checks whether the device/device type passes in date/times that are already shifted to accommodate shift periods such as Daylight Saving Time. If the "Incoming Data Shift Flag" is set on the device, the routine uses this value. Otherwise, it falls back to the value configured on the Device Type. If neither is configured, the routine assumes the date/time is shifted to accommodate shift periods such as Daylight Saving Time.

This algorithm determines the source time zone as follows:

1. The time zone element in the "preVEE" group,
2. The translated version of the external time zone in the "preVEE" group
3. The Service Point time zone, the Device Configuration time zone
4. The Measuring Component time zone
5. The installation time zone.

The target time zone is set to the installation time zone.

See the Detailed Description for the "Shift Event Date/Times to Standard" algorithm (D1-SHEVTDTTM) for more information about this process.

Reader Remarks

Reader remarks are a specific type of device event used to capture and/or record specific events or circumstances encountered when a meter reader is manually reading scalar meters. Reader remarks are submitted with scalar initial measurements when received from a head-end system or meter read collection system. Reader remarks are NOT uploaded along with other device events. Reader remarks are ALWAYS associated with a scalar initial measurement. When a reader remark is created, the reader remark type included in the initial measurement determines the type of reader remark created. See **Reader Remarks and Scalar Measurements** in **Chapter 6: Measurement Data** for more information about reader remarks.

Understanding Service Tasks

This section provides an overview of service tasks and how they are used in Oracle Utilities Meter Data Management.

Service tasks are task-related records, such as tasks performed by users of other Oracle Utilities applications, such as Oracle Utilities Customer Self Service. Service tasks are also used to capture instances when service investigation might be required at a service point based on user-specified circumstances.

- **Self-Service Meter Reads**
- **Service Issue Monitors and Service Investigative Orders**

Self-Service Meter Reads

Self service meter read service tasks represent meter reads created by users of the Oracle Utilities Customer Self Service application. Service tasks of this type reference details specific to the meter and meter read submitted:

- **Service Task User ID:** the user ID of the user who created the service task.
- **Email Address:** the email address of the user who created the service task.
- **IP Address:** The IP address of the server hosting the application used to create the service task.
- **Usage Subscription:** The usage subscription related to the service point for which the meter read was created.
- **Service Point:** The service point for which the meter read was created
- **Device Configuration:** The device configuration for the service point for which the meter read was created.
- **Meter Read Details:** details of the meter read, including read sequence, measuring component, reading, and initial measurement created from the meter reading.

Service Issue Monitors and Service Investigative Orders

Service Issue Monitors are service tasks that analyze service points to determine if service is needed. If service is determined to be needed, the Service Issue Monitor creates a Service Investigative Order.

Device events, VEE exceptions, and failed smart meter commands can trigger the creation of a service issue monitor (the type of service issue monitor created is based on the Service Issue Monitor Type specified on the device event type, exception type, or activity type). Once created, service issue monitors analyze the service point where the device associated with the device event, VEE exception, or failed command, based on evaluation criteria specified on the service issue monitor's type. If the criteria are met (in other words, if a specified number of command failures, device events, or VEE exceptions are found for the service point), the service issue monitor creates a service investigative order.

Attributes used to define service issue monitors include the following:

- **Service Task Type:** The Service Issue Monitor Type upon which the Service Issue Monitor is based.
- **Status:** The current status of the Service Issue Monitor. Can be "Pending", "Approval In Progress", "Processed", or "Discarded".
- **Service Point:** The service point at which the event that triggered the creation of the Service Issue Monitor occurred.

- **VEE Exception ID:** The ID of the VEE exception that triggered the creation of the Service Issue Monitor (if applicable).
- **Device Event ID:** The ID of the device event that triggered the creation of the Service Issue Monitor (if applicable).
- **Initiating Command:** The failed command that triggered the creation of the Service Issue Monitor (if applicable).
- **Resulting Activity:** The Service Investigative Order activity created as a result of the Service Issue Monitor.
- **Events:** Details of the events that triggered the creation of the Service Investigative Order created by the Service Issue Monitor.
 - **Sequence:** The order in which the event occurred.
 - **Event Date/Time:** The date and time at which the event occurred.
 - **Events:** The information string for the event.

Service Investigative Orders

Service investigative orders are activities created by a service issue monitor when a specified set of events have occurred at a service point. The type of activity created by the service issue monitor is defined on the service issue monitor's type.

Service issue monitors are often configured to create field activities that are in turn sent to an external field work system, such as Oracle Utilities Mobile Workforce Management.

Understanding Service Order Monitor Creation and Processing

This section provides an overview of the process that takes place when service issue monitors are created. For each step in the process, the table below provides a brief description of the processing that takes place, and lists the specific objects used by the Service and Measurement Data Foundation

Note that the process outlined below has been simplified for illustrative purposes, and does not reference every step performed in this process.

Step	Process	Service and Measurement Data Foundation Objects
1.	One (or more) of the following takes place: <ul style="list-style-type: none"> • A device event or reader remark is received, whose type specifies a Service Issues Monitor Type • An exception whose type specifies a Service Issues Monitor Type is generated during VEE processing • A smart meter command fails whose type specifies a Service Issues Monitor Type 	All base package device event types support defining a Service Issue Monitor Type. Base package exception types support defining a Service Issue Monitor Type (these must reference the D2-VEEExceptionServiceMonitor VEE Exception Business Object). Base package command activity types support defining a Service Issue Monitor Type.

Step	Process	Service and Measurement Data Foundation Objects
2.	<p>For device events, an Enter algorithm on the Processed state creates the service issue monitor, based on the Service Issue Monitor Type specified for the device event type.</p> <p>For reader remarks, an Enter algorithm on the Executed state creates the service issue monitor:</p>	<p>Device Event Business Object: D1-DeviceEvent Algorithm: D1-DVCEVTSIM (Create Service Issue Monitor from Device Event)</p> <p>Reader Remark Business Object: D1-ReaderRemark Algorithm: D1-DVCEVTSIM (Create Service Issue Monitor from Device Event)</p>
3.	For VEE exceptions, an Enter algorithm on the Processed state creates the service issue monitor, based on the Service Issue Monitor Type specified for the exception type.	<p>VEE Exception Business Object: D2-VEEExceptionServiceMonitor Algorithm: D2-VEEEXCSIM (Create Service Issue Monitor from VEE Exception)</p>
4.	For failed smart meter commands, an Enter algorithm on the Discarded state creates the service issue monitor, based on the Service Issue Monitor Type specified for the exception type.	Algorithm: D1-CFCMDSIM (Create Service Issue Monitor from Failed Command)
5.	<p>Service issue monitors begin in the Pending state. They are transitioned to the Analyze state either manually or via batch processing.</p> <p>An Enter algorithm on the Analyze state analyzes the service point based on the Discard Rules and Evaluation Criteria defined on the service issue monitor's type to determine if a service investigative order should be created or if the service issue monitor should be discarded.</p> <p>The "Discard Rules" on the service issue monitor's type are evaluated to determine if the new service issue monitor should be discarded (new service issue monitors are always discarded when created if an existing service investigative order created by an service issue monitor of the same type exists for the service point.)</p> <p>If a service investigative order should be created, the service monitor moves to the Processed state.</p> <p>If the service issue monitor type was configured to require approval, the service issue monitor moves to the Approval in Progress state.</p> <p>If the service issue monitor should be discarded, it moves to the Discarded state.</p>	<p>Batch Control: F1-STKDF (Service Task Monitor, Deferred)</p> <p>Algorithm: D1-SIM-ANZ (Analyze Service Point for Service Issues)</p>

Step	Process	Service and Measurement Data Foundation Objects
6.	<p>If the Evaluation Criteria on the service issue monitor's type are met, the service issue monitor moves into the Processed state and creates an activity of the type specified in the Service Investigative Order section of the service issue monitor's type.</p> <p>An Enter algorithm on the Processed state creates the activity.</p>	Algorithm: D1-CRE-SIO (Create Service Investigative Order)

Service Task Types

Service task types define properties common to specific types of service tasks.

Service task types represent different types of tasks that can be performed by users of other Oracle Utilities applications, such as Oracle Utilities Customer Self Service or Oracle Utilities Network Management System. Examples of service tasks include self service meter reads, in which users enter their own meter reads via the Customer Self Service application.

Service task types can be defined by the following attributes

- **Service Task Type:** the name of the task type.
- **Service Task Business Object:** the business object instantiated when service tasks of this type are created.
- **Service Task Class:** a category used to service task types for reporting purposes (outage, self-service, etc.).
- Other data based on the specific type of service task (Service Provider, Data Source, Exception Handling, etc.)

About Service Issue Monitor Types

Service issue monitor types are a category of service task types used to define the conditions under which service issue monitors are created. Service issue monitors monitor and analyze service points to determine when service is needed.

Service monitor issue types can be defined by the following attributes:

- **Related Transaction BO:** The business object used to create Service Issue Monitors when the evaluation criteria is met.
- **Service Task Class:** The class of service task. For Service Issue Monitor Types, this should be set to "Service Issue Monitor".
- **Approval Required:** Specifies whether or not approval is required before creating a Service Investigative Order based on this Service Issue Monitor Type.
- **Evaluation Criteria:** Defines the criteria used to determine if a Service Investigative Order should be created. Service Investigative Orders are created if a specified number of command failures, device events, or VEE exceptions are found for the service point. Evaluation criteria are defined by the following:
 - **Sequence:** The order in which the criteria is evaluated.
 - **Evaluation Criteria Relationship:** The relationship between this criteria and other criteria (based on sequence). Valid options are "And" and "Or". If set to "And", a Service Investigative Order is only created if this criteria and all other "And" criteria are met. If set to "Or", a Service Investigative Order is created if this criteria or any other "Or" criteria are met.

- **Service Issue Monitor Evaluation Types:** The type of evaluation to perform for this criteria. This specifies the type of issue to search for. Valid options include Command Failure, Device Event, or VEE Exception.
- **Evaluation Details:** Specific details for the evaluation criteria, based on the evaluation type:
 - **Command Failure:** One or more command (activity) types that indicate a Service Investigative Order should be created
 - **Device Event:** A device event category and one or more device types that indicate a Service Investigative Order should be created
 - **VEE Exception:** The VEE exception type that indicates a Service Investigative Order should be created
- **Number of Occurrences:** The number of occurrences of the command failure, device event, or VEE exception that must occur before a Service Investigative Order is created.
- **Number of Days Back:** The number of days in the past to check for other instances of the command failure, device event, or VEE exception.
- **Discard Rules:** Defines rules for discarding new Service Issue Monitors based on existing Service Investigative Orders. New Service Issue Monitors are always discarded when created if an existing Service Investigative Order created by an Service Issue Monitor of the same type exists for the service point. Discard rules are defined by the following:
 - **If Existing SIO Found with Different SIM Type:** A flag that indicates if the current Service Issue Monitor should be discarded if an outstanding Service Investigating Order of a different type is found.
 - **If Completed SIO Found:** A flag that indicates if the current Service Issue Monitor should be discarded if a completed Service Investigative Order created from a Service Issue Monitor of the same type is found. If Set to "Yes" the "If Existing SIO Found with Different SIM Type" is also evaluated to determine whether or not to discard the Service Issue Monitor if an outstanding Service Investigating Order of a different type is found.
 - **Number of Days Back:** The number of days in the past to check for existing Service Investigative Orders when determining whether or not to discard the Service Issue Monitor.
- **Service Investigative Order:** Defines the type of Service Investigative Order to create if the evaluation criteria are met.
 - **Service Investigative Order Type:** The activity type for activities created when the evaluation criteria are met
 - **Field Task Type:** Specifies the type of field activity. Used only if/when the Service Investigative Order Type is a field activity.

Activities in Detail

This section provides details concerning the activity objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom activity objects you create as part of your implementation. This section includes:

- A description of the D1-ACTIVITY maintenance object
- Lists of the base package activity business objects
- Details concerning activity-specific configuration options
- A sample activity business object (D1-DeviceWithDurationActivity)

Maintenance Object - D1-ACTIVITY

Activity business objects use the D1-ACTIVITY maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-ACTIVITY
Description	Activity
Service Name	D1-ACTIVITY (Activity Maintenance)
Tables	<ul style="list-style-type: none">• D1_ACTIVITY (Activity) - Primary• D1_ACTIVITY_CHAR (Activity Characteristics) - Child• D1_ACTIVITY_IDENTIFIER (Activity Identifier) - Child• D1_ACTIVITY_LOG (Activity Log) - Child• D1_ACTIVITY_LOG_PARM (Activity Log Parameter) - Child• D1_ACTIVITY_REL (Activity Relationship) - Child• D1_ACTIVITY_REL_OBJ (Activity Related Object) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Activity Business Objects

The Service and Measurement Data Foundation base package includes the following activity business objects:

Business Object Name	Description
D1-DeviceEventNotifSuppress	Device Event Notification Suppression Instances of this business object represent a suppression of device event notifications to a particular service provider (an external system).

Business Object Name	Description
D1-DeviceWithDurationActivity	Outage Activity with Duration Instances of this business object represent individual outage activities (triggered by outage device events) in the system.
D1-DeviceWithDurationActParent	Device Activity with Duration Parent (used a parent for device activity with duration business objects)
D1-MeterReadDownloadActivity	Meter Read Download Activity Instances of this business object represent individual meter read downloads in the system.
D1-RelMCREAbstract	Related MC Consumption Sync Abstract (used as a parent business object for interval/scalar related measuring component synchronization business objects)
D1-RelMCREInterval	Related MC Consumption Sync - Interval Instances of this business object represent individual interval consumption synchronization activities in the system.
D1-RelMCREScalar	Related MC Consumption Sync - Scalar Instances of this business object represent individual scalar consumption synchronization activities in the system.
D1-UTCorrectnPrcessrActivity	Usage Transaction Correction Processor Instances of this business object are created via Initial Measurement Data (IMD) processing when the IMD being processed may have resulted in a change to one or more usage transactions for usage subscriptions linked to the measuring component's device via one of its service points.

The Service and Measurement Data Foundation base package includes the following “lite” activity business objects:

Business Object Name	Description
D1-ActivityBIBOToRead	Activity BI BO To Read
D1-ActivityLite	Activity LITE
D1-ActivityParentLite	Activity Parent LITE
D1-SuppressionLite	Suppression LITE

Example Activity - D1-DeviceWithDurationActivity

The table below lists the details of the D1-DeviceWithDurationActivity activity business object.

Option	Description
Business Object	D1-DeviceWithDurationActivity
Description	Outage Activity with Duration
Maintenance Object	D1-ACTIVITY (Activity)
Parent Business Object	D1-DeviceWithDurationActParent
Application Service	D1-DVCWITHDURATNBOAS (Device Activity with Duration BO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> Related Administration BO: D1-DeviceWithDurationActType (Device Activity with Duration Type) Display UI Map: D1-DvcWithDuratnActivityDisp (Outage Activity with Duration - Display) Portal Navigation Option: d1ActivityNavOpt (Activity Navigation Option) Display Map Service Script: D1-RtDDADtIs (Device Activity with Duration - Retrieve Details for Display) Maintenance UI Map: D1-DvcWithDuratnActivityMaint (Outage Activity with Duration- Maintenance)
Algorithms (Inherited from parent business object)	<ul style="list-style-type: none"> Information: D1-DDACINFO (Device Activity with Duration Information) Pre-Processing: D1-DEFACTTYP (Determine Activity Type)
Lifecycle (Inherited from parent business object)	<ul style="list-style-type: none"> Pending (Initial) Started (Interim) Ended (Final)

Use the Business Object portal to view additional details concerning this business object.

Meter Data Management Base Package Activity Business Objects

The Oracle Utilities Meter Data Management base package includes the following activity business objects. These are used to create aggregator measuring components. See **Automatic Creation of Aggregator Measuring Components** on page 13-7 for more information.

Business Object Name	Description
D2-ActivityAggDimScanner	Aggregator Creator - Postal/Service Type
D2-MsrmtQDUCAggScanner	Dimension Scanner - Create Sub Scanner
D2-MsrmtQuantityAggScanner	Measurement Quantity Scanner
D2-SubMsrmtQDUCAggScanner	Dimension Scanner - Create Aggregators

A Note About License Restrictions

Other activity business objects beyond those listed above are restricted for use with other Oracle Utilities products, and cannot be used without the proper license.

Device Events in Detail

This section provides details concerning the device event objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom device event objects you create as part of your implementation. This section includes:

- A description of the D1-DVCEVENT maintenance object
- Lists of the base package device event business objects, including “lite” business objects
- A sample device event business object (D1-SmartMeterdeviceEvent)

Maintenance Object - D1-DVCEVENT

Device event business objects use the D1-DVCEVENT maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-DVCEVENT
Description	Device Event
Service Name	D1-DVCEVENT (Device Event Maintenance)
Tables	<ul style="list-style-type: none"> • D1_DVC_EVT (Device Event) - Primary • D1_DVC_EVT_CHAR (Device Event Characteristic) - Child • D1_DVC_EVT_IDENTIFIER (Device Event Identifier) • D1_DVC_EVT_LOG (Device Event Log) - Child • D1_DVC_EVT_LOG_PARM (Device Event Log Parameter) - Child • D1_DVC_EVT_REL_OBJ (Device Event Related Object) - Child
Algorithms	<ul style="list-style-type: none"> • Transition Error: D1-GEN-MOERR (MO Transition Error)

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Device Event Business Objects

The Service and Measurement Data Foundation base package includes the following device event business objects:

Business Object Name	Description
D1-DeviceEvent	Device Event (generic device event business object used as a parent business object for all other device event business objects.
D1-DeviceEventSeeder	Device Event Seeder (used to determine the device event business object to use when creating new device events)

Business Object Name	Description
D1-DeviceEvtComResp	Device Event Communication Response Instances of this business object represent communications created in response to device events
D1-PairedDeviceEvent	Paired Device Event Parent (used a parent business object for first/last paired device event business objects)
D1-PairedEventFirstDeviceEvent	Device Event - Paired Event (First) Instances of this business object represent individual paired event (first) device events in the system.
D1-PairedEventLastDeviceEvent	Device Event - Paired Event (Last) Instances of this business object represent individual paired event (last) device events in the system.
D1-ReaderRemark	Reader Remark Instances of this business object represent individual reader remarks in the system. Note that reader remarks are created via import of initial measurement data.
D1-StandardDeviceEvent	Standard Device Event Instances of this business object represent individual standard device events in the system.

The Service and Measurement Data Foundation base package includes the following “lite” device event business objects:

Business Object Name	Description
D1-DeviceEventBIBOToRead	Device Event BO To Read

Example Device Event - D1-DeviceEvent

The table below lists the details of the D1-DeviceEvent device event business object. Note that this business object is used as a parent business object for other device events business objects.

Option	Description
Business Object	D1-DeviceEvent
Description	Device Event
Maintenance Object	D1-DVCEVENT (Device Event)
Application Service	D1-DEVICEEVENTBOAS (Device Event BO)
Instance Control	Do not allow new instances
Options	None (options are defined for child business objects)

Option	Description
Algorithms (apply to all child business objects)	<ul style="list-style-type: none">Information: D1-DEVTINFO (Device Event Info)Validation: D1-VALDVCEVT (Validate Device Event)Validation: D1-VALDEXEVT (Validate External Event Name)
Lifecycle (apply to all child business objects)	<ul style="list-style-type: none">Pending (Initial)Additional Processing (Interim)Help (Interim)Discarded (Final)Processed (Final)

Use the Business Object portal to view additional details concerning this business object.

Service Tasks in Detail

This section provides details concerning the service task objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom service task objects you create as part of your implementation. This section includes:

- A description of the F1-SVCTASK maintenance object
- Lists of the base package service task business objects, including “lite” business objects
- A sample service task business object (D1-ServiceIssueMonitor)

Maintenance Object - F1-SVCTASK

Device event business objects use the F1-SVCTASK maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	F1-SVCTASK
Description	Service Task
Service Name	F1-SVCTASK (Service Task Maintenance)
Tables	<ul style="list-style-type: none"> • F1_SVC_TASK (Service Task) - Primary • F1_SVC_TASK_CHAR (Service Task Characteristics) - Child • F1_SVC_TASK_LOG (Service Task Log) - Child • F1_SVC_TASK_LOG_PARM (Service Task Log Parameters) - Child • F1_SVC_TASK_REL_OBJ (Service Task Related Objects) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Meter Data Management Base Package Service Task Business Objects

The Meter Data Management base package includes the following service task business objects:

Business Object Name	Description
D1-ServiceIssueMonitor	Service Issue Monitor Instances of this business object represent individual service issue monitors created in the system
D2-RateCompareScenarioRequest	Rate Compare Scenario Request Instances of this business object represent individual requests for rate comparison calculations by users via the Oracle Utilities Self Service application.

Business Object Name	Description
D2-SSMeterReadTaskl	Self-Service Meter Read Instances of this business object represent individual self-service meter reads created by users via the Oracle Utilities Self Service application.

Example Device Event - D1-ServiceIssueMonitor

The table below lists the details of the D1-ServiceIssueMonitor service task business object.

Option	Description
Business Object	D1-ServiceIssueMonitor
Description	Service Issue Monitor
Maintenance Object	F1-SVCTASK (Service Task)
Application Service	D1-SIMBOAS (Service Issue Monitor BO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none">Related Administration BO: D1-ServiceIssueMonitorType (Service Issue Monitor Type)Portal Navigation Option: d1svctdpTabMenu (Service Task)
Algorithms	<ul style="list-style-type: none">Information: D1-SIM-INFO (Service Issue Monitor Information)
Lifecycle	<ul style="list-style-type: none">Pending (Initial)Analyze (Interim)Approval in Progress (Interim)Processing Error (Interim)Processed (Final)Discarded (Final)

Use the Business Object portal to view additional details concerning this business object.

Configuring Activities and Device Event Objects

This section provides high-level overviews of the steps involved in configuring custom activities and device events. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

Note: The procedures below focus on specific configuration tasks and options related to each of the objects described in this chapter, and do not address all the steps involved in creating business objects, UI maps, algorithms, etc. For more information about these subjects, refer to the Oracle Utilities Application Framework documentation.

Configuring Custom Activities

Configuring custom activities involves the following steps:

1. Design the activity business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom activity-related configuration objects required for your business objects, including:
3. Set up admin records that define the activity types you will use in your implementation.

Configuring Custom Device Events

Configuring custom device events involves the following steps:

1. Design the device event business objects you will need to create for your implementation, including the data and processing required for each.
2. Create your device event business objects.

Note: Device event business objects should reference D1-DeviceEvent as their Parent Business Object.

Configuring Service Tasks

This section provides high-level overviews of the steps involved in configuring custom service tasks. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

Note: The procedures below focus on specific configuration tasks and options related to each of the objects described in this chapter, and do not address all the steps involved in creating business objects, UI maps, algorithms, etc. For more information about these subjects, refer to the Oracle Utilities Application Framework documentation.

Configuring Custom Service Tasks

Configuring custom service tasks involves the following steps:

1. Design the service task type business objects you will need to create for your implementation, including the data and processing required for each.
2. Design the service task business objects you will need to create for your implementation, including the data and processing required for each.
3. Create the custom service task-related configuration objects required for your business objects.
4. Set up admin records that define the service task types you will use in your implementation.

Configuring Service Issue Monitors and Service Investigative Orders

Configuring service issue monitors involves the following steps:

1. Define the exception types, device events, and command activities that will trigger the creation of a service issue monitor.
2. Design and create the field activity business object (and its related activity type business object) that will be used to communicate with your field work system.
3. Define an activity type for the field activity type created in step #2.
4. Configure a service monitor issue type for each set of conditions under which a service issue monitor should be created. Specify the following for each:
 - **Evaluation Criteria:** Criteria that determines if a Service Investigative Order should be created. Reference the command activity types, device event category or type, or exception types defined in step #1.
 - **Discard Rules:** Rules for discarding new Service Issue Monitors based on existing Service Investigative Orders.
 - **Service Investigative Order:** The type of activity to create if the evaluation criteria are met. Reference the field activity type you defined in step #3.
5. Reference the service issue monitor type created in step #4 in the exception types, device events, and command activities that will trigger the creation of a service issue monitor.

Chapter 9

Usage Subscriptions

This chapter provides descriptions of usage subscriptions and usage subscription types, including:

- **Understanding Usage Subscriptions**
- **Usage Subscriptions In Detail**
- **Usage Subscription Types In Detail**
- **Configuring Usage Subscriptions and Usage Subscription Types**

Understanding Usage Subscriptions

This section describes usage subscriptions and their role in the usage calculation process.

Usage Subscriptions

Oracle Utilities Meter Data Management can calculate and send bill determinants to other systems, such as a customer information system (CIS), an external billing system, or some other application. Before bill determinants can be calculated, you must first create a usage subscription. A usage subscription can be thought of as an ongoing request to send one or more service points' usage to one or more external systems, and defines the service point's bill determinants are calculated.

Bill determinants (usage) are derived from the final measurements of the measuring components installed at the usage subscription's service points during the calculation period. A service point is linked to measuring components through a install event linked to the measuring components' device configuration.

Usage Subscription Attributes

Attributes used to define usage subscriptions include the following:

- **Master Usage Subscription:** The "master" usage subscription for this "sub" usage subscription (if applicable). "Master" and "sub" usage subscriptions allow for hierarchical relationships between usage subscriptions to represent situations where a subscription is a subordinate to another usage subscription (such as when a utility is issuing a bill for a third-party service).
- **Start and End Date and Times:** The start and end times that define the time period during which the usage subscription is active.
- **Usage Recipient:** The recipient of bill determinants and usage calculated for the usage subscription, based on Valid Service Providers defined by the usage subscription's type.
- **Usage Approved:** A flag that indicates if usage must be approved before being sent to the recipient.
- **Main Contact:** The main contact for the usage subscription.
- **Time Zone:** The time zone in which the usage subscription is active.
- **Factor Overrides:** One or more factors used as override values when calculating bill determinants or usage.
- **Usage Groups:** The usage groups used to calculate bill determinants or usage. Usage groups are effective-dated and can have an expiration date and time. If all usage groups for the usage subscription have expired or none are in effect when calculations are performed, calculations are performed using the **Fallback Usage Group** defined for the usage subscription's type.
- **Fallback Usage Groups:** Defines effective dated fallback usage groups for the current usage subscription type (used in event that no valid usage groups are defined for the usage subscription).

Note: If the Usage Group Determination Override Algorithm field on the usage subscription type is populated, the specified algorithm is used to determine the usage groups to use, and the usage groups defined in this section are ignored.

- **Rate History:** One or more effective-dated Oracle Utilities Customer Care and Billing rates for the usage subscription. These are used when integrating Oracle Utilities Meter Data Management with Oracle Utilities Customer Care and Billing to support dynamic determination of the usage groups to use when calculating usage for the usage subscription.

- **Measuring Components:** One or more measuring components directly related to the usage subscription. Measuring components of this type are most often aggregator measuring components, but can also include profile measuring components.
- **Service Points:** The service points whose measurements are used to calculate usage for the usage subscription.
 - **Start Date/Time:** The start date and time of the relationship between the usage subscription and the service point. All measurements used calculate usage for the usage subscription should have a start date/time later than this date and time.
 - **Override Start Date/Time:** An override date and time used to specify the start date and time when identifying measurements used to calculate usage for the usage subscription. If not specified, the Start Date/Time is used.
 - **Stop Date/Time:** The stop date and time of the relationship between the usage subscription and the service point. All measurements used calculate usage for the usage subscription should have a stop date/time before than this date and time.
 - **Override Stop Date/Time:** An override date and time used to specify the stop date time when identifying measurements used to calculate usage for the usage subscription. If not specified, the Stop Date/Time is used.
 - **Usage:** A flag that indicates how the consumption from this service point is used when calculating usage for the usage subscription. Valid values include:
 - **Add:** The service point's consumption should be added to the consumption of other service points when calculating usage for the usage subscription.
 - **Exclude:** The service point's consumption should be excluded when calculating usage for the usage subscription.
 - **Subtract:** The service point's consumption should be subtracted from the consumption of other service points when calculating usage for the usage subscription.
 - **Use Percent:** The percentage of consumption from this service point that should be used when calculating usage for the usage subscription.

An Aside: No Account Object Exists

Oracle Utilities Meter Data Management (and related meter data products) is not considered the system of record for accounts or usage subscriptions. The customer information system (or some other system) is considered the system of record for this type of information. In order to minimize the amount of data that must be synchronized between systems, account-oriented attributes used by the meter data products are held on usage subscriptions. For example, if an account's ID and its customer class are relevant to usage calculations, each usage subscription must reference both elements. This is an important distinction to keep in mind when creating custom usage subscriptions for your implementation.

Multiple Service Points and Multiple Measuring Components

At any instance in time:

- A usage subscriptions may be linked to multiple service points.
- A service point may be linked to a single device configuration
- A device configuration may have multiple measuring components

The calculation period for bill determinant calculations can span many days and over this period:

- The service points linked to the usage subscription can change (service points can be added and removed)

- The device configurations installed at the service point can change (due to device reconfigurations and meter exchanges)

This means that values for each usage subscription, bill determinants can be calculated using multiple service points and measuring components.

Contacts

Contacts are individuals or business entities with which a company has contact. A contact exists for every individual or business related to a usage subscription. A single usage subscription can have many contacts, and a single contact may be referenced on many different usage subscriptions. Contacts have a 1-to-1 correlation with a "person" in a customer information system (CIS) and the CIS is considered the system of record for contact information.

Service Points and Contacts

As noted in the description of service points, service points can reference contacts. While this is optional, all usage subscriptions must reference at least one contact.

Note: The base-package name search on the 360° Search looks for usage subscription-related contacts. Use the Service Point Query portal to find a service point using a service point-related contact.

Service Providers

As noted in the service point chapter, service providers can be associated with a market and/or the service points in a market. This is optional and typically only set up in deregulated markets.

Usage subscriptions, on the other hand, must reference a service provider. The service provider is used as the identity of the subscribing system. In other words, you must set up a service provider for any system that subscribes to bill determinants.

Usage Subscriptions In Detail

This section provides details concerning the usage subscription objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom usage subscription objects you create as part of your implementation. This section includes:

- A description of the D1-US maintenance object
- Lists of the base package usage subscription business objects, including “lite” business objects
- Details concerning usage subscription-specific configuration options
- A sample usage subscription business object (ID2-UsageSubscription)

Maintenance Object - D1-US

Usage subscription business objects use the D1-US maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-US
Description	Usage Subscription
Service Name	D1-US (Usage Subscription Maintenance)
Tables	<ul style="list-style-type: none"> • D1_US (Usage Subscription) - Primary • D1_US_CHAR (Usage Subscription Characteristics) - Child • D1_US_CONTACT (Usage Subscription - Contact) - Child • D1_US_DYN_OPT (Usage Subscription - Dynamic Option) - Child • D1_US_DYN_OPT_OVRD (Usage Subscription - Dynamic Option Override) - Child • D1_US_FACTOR_OVRD (Usage Subscription Factor Override) - Child • D1_US_IDENTIFIER (Usage Subscription Identifier) - Child • D1_US_LOG (Usage Subscription Log) - Child • D1_US_LOG_PARM (Usage Subscription Log Parameters) - Child • D1_US_MC (Usage Subscription - Measuring Component) • D1_US_REL (Usage Subscription Relationship) - Child • D1_US_SP (Usage Subscription - Service Point) - Child • D1_US_SP_CHAR (Usage Subscription - SP Characteristics) - Child • D1_US_USG_GRP (Usage Subscription - Usage Group) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Usage Subscription Business Objects

The Service and Measurement Data Foundation base package includes the following "lite" usage subscription business objects:

Business Object Name	Description
D1-USLITE	Usage Subscription LITE
D1-UsageSubscriptionParentLITE	Usage Subscription Parent LITE

Meter Data Management Base Package Usage Subscription Business Objects

The Oracle Utilities Meter Data Management base package includes the following usage subscription business objects:

Business Object Name	Description
D2-UsageSubscription	Usage Subscription Instances of this business object represent individual usage subscriptions defined in the system.

The Oracle Utilities Meter Data Management base package includes the following "lite" usage subscription business objects:

Business Object Name	Description
D2-MasterSubUSLite	Master and Sub Usage Subscription LITE (used to retrieve the usage subscription's sub usage subscriptions)
D2-USMainContact	Usage Subscription's Main Contact LITE (used to retrieve the usage subscription's main contact)
D2-USMainContactLITE	Usage Subscription Main Contact LITE (used to retrieve the usage subscription's main contact)
D2-UsageSubscription-Ext-Char	Usage Subscription LITE
D2-UsageSubscriptionLITE	Usage Subscription LITE
D2-UsageSubscriptionParentLITE	Usage Subscription Parent LITE
D2-UsgTranProUsgSub	Usage Subscription Transaction Processing (describes the structure and rules applicable to Usage Subscription Transaction)
D2-USSPLite	Usage Subscription Service Point Lite

The Oracle Utilities Meter Data Management base package includes the following additional usage subscription business objects:

Business Object Name	Description
D2-SynchronizationAddUS	US Synchronization Add (used when adding a new usage subscription as a result of a data synchronization request)

Example Usage Subscription - D2-UsageSubscription

The table below lists the details of the D2-UsageSubscription usage subscription business object.

Option/Field	Description
Business Object	D2-UsageSubscription
Description	Usage Subscription
Maintenance Object	D1-US (Usage Subscription)
Application Service	D2-USBOAS (Usage Subscription BO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> Synchronization Add BO: D2-SynchronizationAddUS (US Synchronization Add) Display UI Map: D2-USDisplay (Usage Subscription - Display) Portal Navigation Option: d1usmTabMenu (Usage Subscription) Display Map Service Script: D2-USRetDtls (Usage Subscription Retrieve Details for Display) Maintenance UI Map: D2-USMaint (Usage Subscription - Maintenance)
Algorithms	<ul style="list-style-type: none"> Information: D2-USINFO (Usage Subscription Information) Pre-Processing: D1-DEFTIMZON (Default Time Zone value based on Installation Option) Validation: D2-USSPDTVAL (Validate Start and Stop Dates of Usage Subscription's Service Points) Validation: D2-USUGDTVAL (Validate Effectivity Period of Usage Subscription's Usage Groups) Validation: D2-USFOVAL (Validate the Start and Stop Dates of US' Factor Overrides) Validation: D1-VALTIMZON (Validates BO Time Zone value against Installation Option) Validation: D2-VAL-SUBUS (Validate Sub Usage Subscription)
Lifecycle	<ul style="list-style-type: none"> Active (Initial) Inactive (Final)

Use the Business Object portal to view additional details concerning this business object.

Usage Subscription Types In Detail

This section provides details concerning the usage subscription type objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom usage subscription type objects you create as part of your implementation. This section includes:

- A description of the D1-USTYPE maintenance object
- Lists of the base package usage subscription type business objects, including “lite” business objects
- Details concerning usage subscription type-specific configuration options
- A sample usage subscription type business object (D2-UsageSubscriptionType)

Maintenance Object - D1-USTYPE

Usage subscription type business objects use the D1-USTYPE maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-USTYPE
Description	Usage Subscription Type
Service Name	D1-USTYPE (Usage Subscription Type Maintenance)
Tables	<ul style="list-style-type: none">• D1_US_TYPE (Usage Subscription Type) - Primary• D1_US_TYPE_CHAR (Usage Subscription Type Characteristics) - Child• D1_US_TYPE_FB_USG_GRP (Usage Subscription Type - Fallback Usg Grp) - Child• D1_US_TYPE_L (Usage Subscription Type - Language) - Child• D1-US_TYPE_VAL_DYN_OPT_TYPE (Usage Subscr Type - Dynamic Option Type• D1_US_TYPE_VAL_SPR (Usage Subscription Type - Valid Service Provider) - Child• D1_US_TYPE_SP_TYPE (Usage Subscription Type - Valid SP Type) - Child• D1_US_TYPE_VAL_USG_GRP (Usage Subscription Type - Valid Usage Group) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Meter Data Management Base Package Usage Subscription Type Business Objects

The Oracle Utilities Meter Data Management base package includes the following usage subscription type business objects:

Business Object Name	Description
D2-UsageSubscriptionType	Usage Subscription Type Instances of this business object represent individual usage subscription types defined in the system.

The Oracle Utilities Service and Measurement Data Foundation and Oracle Utilities Meter Data Management base packages include the following “lite” usage subscription type business objects:

Business Object Name	Description
D1-UsageSubTypeParentLite	Usage Subscription Type Parent - LITE
D1-UsageSubscriptionTypeLite	Usage Subscription Type - LITE
D2-USTypeParentLITE	Usage Subscription Type Parent LITE
D2-UsageSubsTypeParentLITE	Usage Subscription Type Parent - LITE
D2-UsageSubscriptionTypeLITE	Usage Subscription Type LITE
D2-UsgTranProSubTyp	Usage Subscription Type Transaction LITE (describes the structure and rules applicable to Usage Subscription Type Transaction)

The Oracle Utilities Service and Measurement Data Foundation base package includes the following additional usage subscription type business objects:

Business Object Name	Description
D1-UsageSubscrTypeBundlingAddBO	Bundling Add BO for Usage Subscription Type
D1-UsageSubscrTypePhysicalBO	Physical BO for Usage Subscription Type

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by usage subscription type business objects.

Other Options

Usage subscription types define many attributes of the usage subscriptions of that type. These options are specified when creating usage subscriptions based on a usage subscription business object, and include the following:

Valid Service Point Types

These define the service point types considered valid for usage subscriptions of this type.

Valid Service Providers

These define the service providers considered valid for usage subscriptions of this type.

Usage Groups (Valid and Fallback)

Usage Groups define the usage rules to be applied to initial measurement data for usage subscriptions of this type.

- **Valid Usage Groups:** These define the usage groups considered valid for measuring components of this type.
- **Fallback Usage Groups:** These define the usage groups that can be used with all usage subscriptions of this type in situations where the usage groups defined for the usage subscriptions are not in effect. at the time usage is to be calculated Fallback usage groups have effective dates which define the point in time after which they are considered in effect.

Example Usage Subscription Type - D2-UsageSubscriptionType

The table below lists the details of the D2-UsageSubscription usage subscription type business object.

Option/Field	Description
Business Object	D2-UsageSubscriptionType
Description	Usage Subscription Type
Maintenance Object	D2-USTYPE (Usage Subscription Type)
Application Service	D2-USTYPE (Usage Subscription Type MO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none">• Display UI Map: D2-USTypeDisplay (Usage Subscription Type - Display)• Portal Navigation Option: d1ustypeTabMenu (Usage Subscription Type)• Maintenance UI Map: D2-USTypeMaint (Usage Subscription Type - Maintenance)

Use the Business Object portal to view additional details concerning this business object.

Configuring Usage Subscriptions and Usage Subscription Types

This section provides high-level overviews of the steps involved in configuring custom usage subscriptions and usage subscription types. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

Note: The procedures below focus on specific configuration tasks and options related to each of the objects described in this chapter, and do not address all the steps involved in creating business objects, UI maps, algorithms, etc. For more information about these subjects, refer to the Oracle Utilities Application Framework documentation.

Configuring Custom Usage Subscriptions

Configuring custom usage subscriptions involves the following steps:

1. Design the usage subscription business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom usage subscription-related configuration objects required for your business objects.
3. Create your usage subscription business objects, referencing the configuration objects created above as appropriate.

Configuring Custom Usage Subscription Types

Configuring custom usage subscription types involves the following steps:

1. Design the usage subscription type business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom usage subscription type-related configuration objects required for your business objects, including:

Options: Create data as appropriate for the following options used when creating usage subscription types:

- Valid Service Point Types
 - Valid Service Providers
 - Usage Groups (Valid and Fallback)
3. Create your usage subscription type business objects, referencing the configuration objects created above as appropriate.
 4. Set up admin records that define the usage subscription types you will use in your implementation.

Chapter 10

Usage Groups and Usage Rules

This chapter provides descriptions of usage groups and rules, including:

- **Understanding Usage Groups and Rules**
- **Usage Groups In Detail**
- **Usage Rules In Detail**
- **Usage Eligibility Criteria In Detail**
- **Configuring Usage Rules, Groups, and Eligibility Criteria**

Understanding Usage Groups and Rules

This section describes the usage calculation process used Oracle Utilities Service and Measurement Data Foundation and Oracle Utilities Meter Data Management, including descriptions of usage groups and rules.

The Usage Calculation Process

Oracle Utilities Meter Data Management can calculate and publish usage calculated from measurement data to service providers on an ongoing basis. In addition, external systems can request usage whenever needed. Usage calculations derive a usage transaction's usage quantities using the measurements linked to a usage subscription's service points.

The usage calculation engine and process is very similar to the VEE engine in that it is driven by configurable rules. These rules calculate a usage transaction's usage (also known as bill determinants). Usage calculation rules can also be configured to validate the usage that was calculated by earlier rules. If problems are found, exceptions are created and the usage transaction is not finalized.

Most requests for usage result in the creation of a usage transaction, but it is possible for an external application to invoke the usage calculation engine real-time. In other words, usage can be retrieved for a usage subscription real-time without creating a usage transaction. This technique is only recommended for online requests, not as part of batch processes.

See **Chapter 12: Usage Transactions** for more information about the usage calculation process.

Usage Rules and Usage Groups

The specific usage calculation processing performed on (final) measurement data is defined in individual usage rules, each performing a specific set of calculations. The base package includes rules that calculate common bill determinants including:

- Scalar reads
- Time-of-use consumption (by applying a time-of-use map to an interval channel)
- Interval curves (either real or derived)
- Virtually anything else that can be calculated from the information in the system

The base package contains several usage rules you can use in your implementation, and you can also create your own custom usage rules based on your requirements.

Usage Groups

Usage groups are collections of usage rules that are applied to measurement data. During the usage calculation process, the system executes the usage rules defined in each usage group. The rules within a usage group are defined in a specific sequence, allowing control over the order in which the rules are executed.

Fallback and Valid Usage Groups

Usage groups can be associated to a specific usage subscription, or to a usage subscription type (or both). Usage groups associated to a usage subscription type are considered “fallback” usage groups. In addition, a usage subscription's usage subscription type also defines the usage groups that can be defined for individual usage subscriptions, and that are considered “valid” to override the fallback usage groups. Only the valid usage groups on the usage subscription type can be referenced on an individual usage subscription. An individual usage subscription can have override usage groups. If the usage subscription doesn't have an override usage group for a bill determinant's calculation period, the fallback group defined on the usage subscription type is used.

Effective Dates

Unlike VEE rules, usage rules do not have individual effective dates. Usage groups associated to a usage subscription have effective and expiration dates, which define the date range during which they can be applied to usage calculations for the usage subscription. A usage subscription can have many usage groups where each has a different effective period (and multiple usage groups can be effective during a usage transaction's period). In other words, the entire usage group is effective-dated rather than the individual rules.

Note: In MDM 2.0.0, the system uses the group effective at the start of the calculation period. In MDM 2.0.1, the system calculates separate sets of bill determinants for each group in effect during the usage period.

Eligibility Criteria

Each usage rule may optionally have eligibility criteria that controls if the rule is applied. This feature can greatly reduce the number of usage groups you need to create, because it allows a single usage group to have conditional usage rules based on eligibility criteria (rather than requiring a distinct usage group for every combination of v rules).

For example, you might use eligibility criteria that specifies that a rule is only applied if the customer has solar power (or some other unique characteristic).

The base package contains two usage rules eligibility criteria business objects:

- The “Usage Rule Eligibility Criteria” (D1-UsgruleEligibilityCriteria) business object can be used to define eligibility criteria based on the evaluation of some attribute of the device or measuring component.
- The “Usage Rule Eligibility Criteria - UOM/TOU/SQI Eligibility” (D2-VerifyUomTouSqiMCExistForUs) business object can be used to specifically validate whether the usage subscription is linked (via service point / install event / device configuration) to a measuring component with a primary measurement value with a particular UOM/TOU/SQI. For example, rules for calculation of kWh service quantities may apply to all electric commercial customers, and a subset of commercial customers may also require calculations based on kvarh. This eligibility criteria business object can be used to ensure that the rules for kvarh calculations are only executed for those usage subscriptions with associated measuring components that measure kvarh.

Referred Usage Group Rules - Reusing Groups Of Rules

Just like VEE rules, a usage rule can reference a different usage group, so commonly used rules can be encapsulated in reusable usage groups. A usage rule that executes the rules in a referenced usage group is called an Execute Usage Group rule. Rules of this type can have effective eligibility criteria, just like all usage rules.

Execute Usage Group rules can be “nested.” That is, a group executed by a Execute Usage Group rule can, in turn, execute the rules in another group, and so on.

Using Factors For Variables

A situation common in many implementations involves converting one unit of measure (UOM) to another. However, the conversion factor used in conversions of this can differ based on many different types of criteria, such as the location of the service point or other characteristics. This sort of calculation can be implemented as a usage rule that accumulates consumption for one UOM and converts the consumption to a different UOM by applying a factor to it.

Factors used for this purpose have a Factor Class of “Number,” and use some unique rules:

- Number factors reference a characteristic type (with pre-defined values).
- Number factors reference an algorithm that retrieves or derives the value of the characteristic type at runtime.

- Factor values for a Number factor are effective-dated pairings of a characteristic value and a corresponding value. Because these pairings are effective-dated, the value returned from the factor can change over time for each characteristic value

At run time, the rule retrieves / derives the characteristic value for the factor's characteristic type and then finds the value associated with the respective characteristic value.

Factors can be related to any real or dynamic attribute, so rules of this type are very flexible. For example:

- **Real Attribute:** you could create a rule that retrieves a specific value based on the location of a service point.
- **Dynamic Attribute:** you could create a rule that retrieves a percentage value based on the amount the customer conserved as compared to the same period in the prior year, returning one value if the amount conserved is between 5% and 10%, another value if the amount conserved is between 10% and 20%, and yet a third value if the amount conserved is greater than 20%. The amount conserved is dynamically calculated at execution time and is compared to the characteristic values defined for the factor, and returns the appropriate value. In this example, if the amount conserved was anything less than 5%, no percentage value would be returned.

Usage Groups In Detail

This section provides details concerning the usage group objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom usage group objects you create as part of your implementation. This section includes:

- A description of the D1-USGGRP maintenance object
- Lists of the base package usage group business objects, including “lite” business objects
- A sample usage group business object (D1-UsageGroup)

Maintenance Object - D1-USGGRP

Usage group business objects use the D1-USGGRP maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-USGGRP
Description	Usage Group
Service Name	D1-USAGEGRP (Usage Group Maintenance)
Tables	<ul style="list-style-type: none"> • D1_USG_GRP (Usage Group) - Primary • D1_USG_GRP_CHAR (Usage Group Characteristics) - Child • D1_USG_GRP_L (Usage Group Language) - Child • D1_USG_GRP_VAL_DC_TYPE (Usage Group - Valid DC Types) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Meter Data Management Base Package Usage Group Business Objects

The Oracle Utilities Meter Data Management base package includes the following usage group usage business objects:

Business Object Name	Description
D1-UsageGroup	Usage Group Instances of this business object represent individual usage groups defined in the system.

The Oracle Utilities Meter Data Management base package includes the following additional usage group business objects:

Business Object Name	Description
D1-UsageGroupBundlingAddBO	Bundling Add BO for Usage Group
D1-UsageGroupPhysicalBO	Physical BO for Usage Group

Example Usage Group - D1-UsageGroup

The table below lists the details of the D1-UsageGroup device business object.

Option/Field	Description
Business Object	D1-UsageGroup
Description	Usage Group
Maintenance Object	D1-USGGRP (Usage Group)
Application Service	D1-USAGEGRP (Usage Group MO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none">• Display UI Map: D1-UsageGroupDisplay (Usage Group - Display)• Portal Navigation Option: d1usggrpTabMenu (Usage Group)• Maintenance UI Map: D1-UsageGroupMaint (Usage Group - Maintenance)

Use the Business Object portal to view additional details concerning this business object.

Usage Rules In Detail

This section provides details concerning the usage rule objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom usage rule objects you create as part of your implementation. This section includes:

- A description of the D1-USGRULE maintenance object
- Lists of the base package usage rule business objects, including “lite” business objects
- Details concerning usage rule-specific configuration options
- A sample usage rule business object (D2-ApplyMathInt)
- A list of base package usage rules, including the algorithm / algorithm type and a brief description of each

Maintenance Object - D1-USGRULE

Device business objects use the D1-USGRULE maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-USGRULE
Description	Usage Rule
Service Name	D1-USAGERULE (Usage Rule Maintenance)
Tables	<ul style="list-style-type: none"> • D1_USG_RULE (usage Rule) - Primary • D1_USG_RULE_CHAR (usage Rule Characteristics) - Child • D1_USG_RULE_L (usage Rule Language) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Usage Rule Business Objects

The Service and Measurement Data Foundation base package includes the following usage rule business objects:

Business Object Name	Description
D1-GenericUsageRule	Generic Usage Rule (generic usage rule business object used as a parent business object for all other usage rule business objects)
D1-UsageRuleReferredUsageGroup	Execute Usage Group Instances of this business object represent individual Execute Usage Group usage rules defined in the system.

The Service and Measurement Data Foundation base package includes the following “lite” usage rule business objects:

Business Object Name	Description
D1-UsageRuleExeReSequenceLite	Usage Rule Execution Resequencing LITE
D1-UsageRuleParentLITE	Usage Rule Parent LITE

The Service and Measurement Data Foundation base package includes the following additional usage rule business objects:

Business Object Name	Description
D1-UsageRuleBundlingAddBO	Bundling Add BO for VEE Rule
D1-UsageRulePhysicalBO	Physical BO for VEE Rule

Meter Data Management Base Package Usage Rule Business Objects

The Oracle Utilities Meter Data Management base package includes the following usage rule business objects:

Business Object Name	Description
D2-ApplyMathInt	Apply Math (Interval Data)
D2-CalFrequentlyReadScalar	Daily Scalar Usage Rule
D2-ChkExistenceofInstalledDvc	Check Existence of Installed Device
D2-GetIntervalData	Get Interval Data
D2-GetItemCountsConsumption	Get Item Counts and Consumption
D2-GetScalar	Get Scalar Details
D2-GetTOUUsage	Get TOU Mapped Usage
D2-IntervalTierCalculation	Interval Tier Calculation
D2-Math	Vector and Service Quantity Math
D2-ProfileAccumulation	Profile Accumulation
D2-ValAgainstTol	Validate Against Tolerance

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by usage rule business objects.

System Events

Usage rule business objects can make use of the following system events:

- **Apply Usage Rule:** This system event defines the algorithm to use when executing the usage rule.
- **Validation:** This system event defines the algorithm to use to validate the measurement data used by the usage rule.

Other Options

Usage rules use various parameters and properties. These options are specified when creating usage rules based on a usage rule business object, and include the following:

Utility Usage Rules

The Service and Measurement Data Foundation includes one "utility" base package usage rule type that can be used when configuring usage groups and rules. This section outlines the configuration options you need to configure before you can create rules of these types.

Execute Usage Group: Execute Usage Group rules reference a usage group. You must create the usage group to reference before can create rules of this type.

Example Usage Rule - D2-ApplyMathInt

The table below lists the details of the D2-ApplyMathInt device business object.

Option/Field	Description
Business Object	D2-ApplyMathInt
Description	Apply Math (Interval Data)
Maintenance Object	D1-USGRULE (Usage Rule)
Parent Business Object	D1-GenericUsageRule (Generic Usage Rule)
Lifecycle Business Object	Generic Usage Rule
Application Service	D1-USAGERULE (Usage Rule MO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> Display UI Map: D2-ApplyMathIntDisplay (Usage Rule-Apply Math (Interval Data) Display) Maintenance UI Map: D2-ApplyMathIntMaint (Usage Rule-Apply Math (Int) - Maintenance)
Algorithms	<ul style="list-style-type: none"> Apply Usage Rule: D2-APPMATHIN Apply Math (interval data) Validation: D2-VALAPPMTH (Apply Math on Interval Data - Validation)

Use the Business Object portal to view additional details concerning this business object.

Base Package Usage Rules Summary

The following table lists the back package usage rules. Each of these usage rules is provided as a business object and corresponding algorithm/algorithm type.

Usage Rule	Business Object	Algorithm / Algorithm Type	Description
Generic Usage Rule	D1-GenericUsageRule	N/A	Parent usage rule for all other usage rules, contains basic usage rule schema elements
Apply Math (Interval Data)	D2-ApplyMathInt	D2-APPMATHIN	A usage rule used to summarize interval measurements and apply a mathematical formula against the results to derive a usage quantity.
Check Existence of Installed Device	D2-ChkExistenceofInstalledDvc	D2-CHKEXTDVC	A usage rule used to check for the existence of a device installed on the usage subscription's service point for the usage period. It also checks if multi-items, that are effective during the usage period, exist for the usage period.
Daily Scalar Usage Rule	D2-CalFrequentlyReadScalar	D2-GETFRESCR	A usage rule used to calculate usage of daily scalar measuring components of device installed on a service point associated with a usage subscription.
Get Interval Data	D2-GetIntervalData	D2-GETINTDAT	A usage rule used to get interval data for a measuring component and date range.
Get Items Counts and Consumption	D2-GetItemCountsConsumption	D2-GETITEMCC	A usage rule used to find item-based and multi-item-based service points linked to the usage subscription for the current usage transaction, summarizes the item counts by item type and service point, and calculates item-based consumption.
Get Scalar Details	D2-GetScalar	D2-GETSCALAR	A usage rule used to assemble scalar readings and measurements (consumption).
Get TOU Mapped Usage	D2-GetTOUUsage	D2-GETTOUSG	A usage rule used to summarize interval measurements into TOU buckets based on a TOU map (where the TOU map is created based on the schedule defined within a TOU map template).
Interval Tier Calculation	D2-IntervalTierCalculation	D2-GETINTIER	A usage rule used to calculate the difference between a source and reference vector. It then breaks that difference into one to many positive or negative tiers.
Profile Accumulation	D2-ProfileAccumulation	D2-DYNPRFLAC	A usage rule used to manipulate a customer's interval data by adding other vectors to it. Those other vectors are derived from a list of profile factors and corresponding characteristic values stored in a list on the usage transaction.

Usage Rule	Business Object	Algorithm / Algorithm Type	Description
Vector and Service Quantity Math	D2-Math	D2-MATH	A usage rule used to derive a curve of interval values given a formula, apply TOU mapping to a derived curve, and perform mathematical operations on Usage Transaction Service Quantity entries.
Validate Against Tolerance	D2-ValAgainstTol	D2-VALUSGTOL	A usage rule used to validate calculated usage against a specified tolerance.

Use the Business Object portal and Algorithm portal (or Application Viewer) to view additional details about these usage rules.

Usage Eligibility Criteria In Detail

This section provides details concerning the usage rule eligibility criteria objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom usage rule eligibility criteria objects you create as part of your implementation. This section includes:

- A description of the D1-USGRLELIG maintenance object
- Lists of the base package usage rule eligibility criteria business objects, including “lite” business objects
- Details concerning usage eligibility criteria-specific configuration options
- A sample usage rule eligibility criteria business object (D1-UsgRuleEligibilityCriteria)

Maintenance Object - D1-USGRLELIG

usage eligibility criteria business objects use the D1-USGRLELIG maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-USGRLELIG
Description	Usage Rule Eligibility Criteria
Service Name	D1-USGRLELIG (Usage Eligibility Criteria Maintenance)
Tables	<ul style="list-style-type: none"> • D1_USG_RULE_ELIG_CRIT (Usage Rule Eligibility Criteria) - Primary • D1_USG_RULE_ELIG_CRIT_CHAR (Usage Rule EC Characteristics) - Child • D1_USG_RULE_ELIG_CRIT_L (Usage Rule Eligibility Criteria Language) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Service and Measurement Data Foundation Base Package Usage Rule Eligibility Criteria Business Objects

The Service and Measurement Data Foundation base package includes the following usage rule eligibility criteria business objects:

Business Object Name	Description
D1-UsgRuleEligibilityCriteria	Usage Rule Eligibility Criteria Instances of this business object represent individual usage rule eligibility criteria defined in the system.

The Service and Measurement Data Foundation base package includes the following additional usage rule eligibility criteria business objects:

Business Object Name	Description
D1-UsageRuleEligCritBundlingAddBO	Bundling Add BO for VEE Eligibility Criteria

Business Object Name	Description
D1-UsageRuleEligCritPhysicalBO	Physical BO for VEE Eligibility Criteria

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by usage rule eligibility criteria business objects.

System Events

Usage rule eligibility criteria business objects can make use of the following system events:

- **Apply Usage Rule Eligibility Criteria:** This system event defines the algorithm to use to apply eligibility criteria to a usage rule.

Example Usage Rule Eligibility Criteria - D1-UsgRuleEligibilityCriteria

The table below lists the details of the D1-UsgRuleEligibilityCriteria usage rule eligibility criteria business object.

Option/Field	Description
Business Object	D1-UsgRuleEligibilityCriteria
Description	Usage Rule Eligibility Criteria
Maintenance Object	D1-USGRLELIG (Usage Rule Eligibility Criteria)
Application Service	D1-USGRLELIG (Usage Rule Eligibility Criteria MO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> • Display UI Map: D1-UsageEligibilityCritDisp (Usage Rule Eligibility Criteria Display) • Portal Navigation Option: d1usgrleTabMenu (Usage Rule) • Maintenance UI Map: D1-UsageEligibilityCritMaint (Usage Rule Eligibility Criteria Maintenance)
Algorithms	<ul style="list-style-type: none"> • Apply Usage Rule Eligibility Criteria: D1-ECF-APUSG (Evaluate Criteria Field - Apply Eligibility Criteria)

Use the Business Object portal to view additional details concerning this business object.

Meter Data Management Base Package Usage Eligibility Criteria Business Objects

The Oracle Utilities Meter Data Management base package includes the following usage rule eligibility criteria business objects:

Business Object Name	Description
D2-VerifyUomTouSqlMCExistForUs	Usage Rule Eligibility Criteria - UOM/ TOU/SQL Eligibility Used to apply a usage rule based on whether the usage subscription is linked (via service point / install event / device configuration) to a measuring component with a primary measurement value with a particular UOM/TOU/SQL.

Configuring Usage Rules, Groups, and Eligibility Criteria

This section provides high-level overviews of the steps involved in configuring custom usage groups and rules. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

This section also provides information related to creating instances of the Referred Usage Group “generic utility” usage rule described earlier in this chapter.

Note: The procedures below focus on specific configuration tasks and options related to each of the objects described in this chapter, and do not address all the steps involved in creating business objects, UI maps, algorithms, etc. For more information about these subjects, refer to the Oracle Utilities Application Framework documentation.

Configuring Custom Usage Groups

Configuring custom usage groups involves the following steps:

1. Design the usage group business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom usage group-related configuration objects required for your business objects.
3. Create your usage group business objects, referencing the configuration objects created above as appropriate.

Configuring Custom Usage Rules

Configuring custom usage rules involves the following steps:

1. Design the usage rule business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom usage rule-related configuration objects required for your business objects, including:

System Events: Create algorithms for the following system events:

- Apply Usage Rule

Options: Create data as appropriate for the following options used when creating usage rules:

- Generic Utility Usage Rules: Define the following options used with “generic utility” usage rules
 - Referred Usage Group: Usage groups to be referenced by these rules.

3. Create your usage rule business objects, referencing the configuration objects created above as appropriate.

Note: Usage rule business objects should reference D1-GenericUsageRule as their Parent Business Object.

Configuring Custom Usage Rule Eligibility Criteria

Configuring custom usage eligibility criteria involves the following steps:

1. Design the usage eligibility criteria business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom usage eligibility criteria-related configuration objects required for your business objects, including:

System Events: Create algorithms for the following system events:

- Apply Usage Rule Eligibility Criteria
3. Create your usage eligibility criteria business objects, referencing the configuration objects created above as appropriate.

Creating Execute Usage Group Usage Rules

Use the following procedure to create Execute Usage Group usage rules:

1. Create the usage group to which the rule will belong.
2. Create the usage group that the rule will reference
3. Create the usage rule referencing the group created in the previous step

Creating Number Factors

Use the following procedure to create number factors used in usage rules.

1. Create the Characteristic Type and Values to be used by the factor that will be referenced by the rule.
2. Create the Characteristic Source Algorithm to be used by the factor that will be referenced by the rule.
3. Create the Factor that will be referenced by the rule.
4. Create the Factor Values for the factor, each referencing an effective-dated characteristic value/value pairings.

Chapter 11

TOU Maps and Dynamic Options

This chapter provides descriptions of time of use (TOU) maps and dynamic options. This chapter includes:

- **Understanding Time of Use, TOU Maps, and Dynamic Options**
- **TOU Maps in Detail**
- **Dynamic Options in Detail**
- **Dynamic Option Events in Detail**
- **Configuring TOU Maps and Dynamic Options**

Understanding Time of Use, TOU Maps, and Dynamic Options

This section describes time of use, TOU maps, dynamic options, and dynamic option events, and how they are used in Oracle Utilities Service and Measurement Data Foundation and Oracle Utilities Meter Data Management.

Time of Use

A type of rate used at many utilities is known as a “time of use” rate. For these rates, the price is based on when the usage occurs. For example, prices are typically higher during peak consumption periods (commonly referred to as “On Peak” periods) to encourage lower use during these times. The time range when a given price is applicable is known as a time-of-use (TOU) period.

Time of use periods can (and often do) change during the year.

For example, in the summer, TOU periods might be defined as follows:

Time of Use Period	Time Periods
On Peak	8:00 AM - 9:00 PM Monday - Friday
Off Peak	12:00 AM - 8:00 AM Monday - Friday 9:00 PM - 12:00 AM Monday - Friday 12:00 AM - 12:00 AM Saturday, Sunday, and Holidays

While winter TOU periods might be defined as follows:

Time of Use Period	Time Periods
On Peak	11:00 AM - 5:00 PM Monday - Friday
Off Peak	12:00 AM - 11:00 AM Monday - Friday 5:00 PM - 12:00 AM Monday - Friday 12:00 AM - 12:00 AM Saturday, Sunday, and Holidays

This is an example of a "simple" mass-market time-of-use schedule; commercial and industrial schedules can be much more complex.

Recording Usage for Time of Use Periods

Generally speaking, there are two different types of meters that can be installed at service points that participate in TOU rates:

- Scalar meters with dedicated registers for each TOU period. For this type of meter, the scalar measurements are the TOU consumption and no special derivation is necessary.
- Interval meters that measure how much is consumed each interval. For this type of meter, the interval measurements are used to derive the consumption in each TOU period.

TOU Periods are Used by the User Interface and Usage Calculation Process

While the requirement to aggregate interval consumption into TOU periods is the principal reason behind TOU maps, the system has many zones that display interval consumption using TOU periods.

The TOU concepts described in this chapter are also used when bill determinants are aggregated into TOU periods as part of the usage calculation process.

TOU Maps

Interval consumption is mapped into time-of-use periods using a TOU map. TOU maps define the TOU periods for each interval. TOU map data is structured in a format very similar to final measurement data, with specified intervals of time each designated a TOU period (instead of a measurement value). For example, the table below lists a set of TOU map data and corresponding final measurements for a period of 4 hours on January 1, 2010.

TOU Map: 463524764		Measuring Component ID: 31425364	
Date/Time	TOU Code	Date/Time	Final Consumption
1-Jan-10 3:00pm	ONPEAK	1-Jan-10 3:00pm	14.678
1-Jan-10 4:00pm	ONPEAK	1-Jan-10 4:00pm	29.12
1-Jan-10 5:00pm	ONPEAK	1-Jan-10 5:00pm	12.12
1-Jan-10 6:00pm	OFFPEAK	1-Jan-10 6:00pm	17.16

Any type of final measurement can be mapped to a TOU period, including derived quantities,

TOU Periods and TOU Groups

Some utilities use many TOU periods. For example, a TOU rate might have different prices depending on:

- The season of the year (winter, spring, summer, fall)
- The time of day (peak, off peak, shoulder)
- The actual temperature as compared to the historic average (normal, hotter, colder)

This rate would require 24 TOU periods (4 seasons * 3 time of days * 3 temperature bands). In contrast to this, another rate might have prices that differ only depending on the time of day (peak, off peak, shoulder). This rate would require only 3 TOU periods.

To help in tracking and maintaining TOU periods, you can create TOU groups, which define the TOU periods that can be used on a TOU map.

When TOU data is created for a TOU map, only TOU periods defined on a specified TOU group can be specified.

TOU Map Templates

Every TOU map references a TOU map template that defines the rules for generating TOU data from that TOU map. Specifically, TOU map templates define:

- The TOU group (defines the valid TOU periods for the template) used for the TOU map
- The default TOU period used for periods not explicitly defined. (This means you don't have to specify dates and times for all periods. For example, if your default TOU period is "Off Peak" you only need to define dates and days and times for On Peak or other TOU periods.)
- The specific date ranges, days of the week, and time periods designated for each TOU period.

The system periodically generates TOU map data for TOU maps by interpreting the rules defined template.

Holidays

Many utilities categorize consumption on holidays differently than on the day of week on which the holiday falls. For example, holiday consumption might be categorized as Off-Peak regardless of the day it falls on. TOU map templates can define rules for different TOU periods for holidays by specifying the following:

- A Work Calendar that defines when holidays start and end
- Either:
 - A Holiday TOU period for consumption on holiday
 - A Holiday TOU Map Template that defines the TOU codes to use for different times in the year

TOU Map Template Interval Size

TOU map templates can also specify an interval size (in seconds-per-interval, or SPI). This value specifies the duration of the individual TOU map data records, and also controls the values allowed in the Start and End Times. For example, if a TOU map template sets the interval size at 15 minutes, Start and End times must be in units of the interval size (10:00, 10:15, 10:30, etc.).

A TOU map template can be used to generate TOU map data for TOU maps whose SPI is divisible by the template's SPI. For example, a 60 minute template can be used to generate TOU data for TOU maps with SPIs of 60 minutes, 15 minutes, 5 minutes, etc. This means separate map templates are not needed for every SPI.

TOU Map Types

TOU map types define attributes shared by TOU maps of a given type. Among these are the business object used for TOU maps of the given type, as well as interval size and TOU map templates.

TOU Map Type Interval Size

As noted in the **Chapter 4: Devices, Measuring Components, and, Device Configurations**, a measuring component's measuring component type defines its interval size (or SPI), which in turn controls the times on the measuring component's measurements.

In a similar way, every TOU map references a TOU map type that defines its interval size (and other properties). This SPI controls the times on the TOU map's TOU data.

The SPI of a TOU map must divide evenly into the SPI of any measuring component that uses the map (because the system joins the date/time of the measurement to the date/time of the TOU data). This means that it is possible to use a 15 minute TOU map with a 60 minute measuring component. However, it is not OK to have a 60 minute TOU map used with a 15 minute measuring component because the join will miss 3 out of 4 measurements.

Default and Override TOU Map Templates

While most TOU maps will use the TOU map template defined on the TOU map type, TOU maps also support a fallback/override pattern used in other areas of the system.

- A TOU map's TOU map type defines the default (or "fallback") TOU map template that's used to generate its TOU data.
- A TOU map's type defines the TOU map templates that can be referenced on individual TOU maps to override the "fallback" template.
- An individual TOU map can have an override template. If the TOU map doesn't have an override template, the fallback template defined on the TOU map type is used to generate the map's TOU data.

Dynamic Options and Dynamic Option Events

There are circumstances and conditions during which the rules for creating TOU map data might need to be calculated differently than according to the utility's standard rules. Examples of this might include critical peak periods, curtailment requests, or demand response events. During these types of events, the TOU rules defined for a TOU map must be overridden. This is done through the use of dynamic options, and dynamic option events.

Dynamic options define specific types of events which can impact how TOU map data is generated. Using the examples listed above, you might create dynamic options such as the following:

- Critical Peak Period (CPP)
- Curtailment (CURTAIL)
- Demand Response Event (DR_EVENT)

Dynamic option events define the specific periods of time during which a dynamic option is in effect. For example, if the utility identifies the period between 10:00 AM and 2:00 PM on August 2 as a critical peak period, a dynamic option event for this might look like this:

- **Dynamic Option ID:** Critical Peak Period
- **Start Date/Time:** 08/02/2010 10:00 AM
- **Stop Date/Time:** 08/02/2010 2:00 PM

A dynamic option may have many dynamic option events over time.

Dynamic Options and TOU Maps

To apply a dynamic option (and one or more of its related dynamic option events), you reference the dynamic option on a TOU map, along with a corresponding “dynamic” TOU map to be used during the dynamic option event.

To continue the example from above, for Critical Peak Periods you might add a new TOU period (called “Critical Peak”), and create a new TOU map that is the same as your standard TOU map, but that also includes your Critical Peak TOU period. For example, in the summer your new set of TOU periods might be defined as follows:

Time of Use Period	Time Periods
Critical Peak	10:00 AM - 2:00 PM Monday - Friday
On Peak	8:00 AM - 9:00 PM Monday - Friday
Off Peak	12:00 AM - 8:00 AM Monday - Friday 9:00 PM - 12:00 AM Monday - Friday 12:00 AM - 12:00 AM Saturday, Sunday, and Holidays

If a dynamic option event occurs during a usage transaction period, the standard TOU map is overridden with the “dynamic” TOU map. In our example, a usage transaction period that includes August 2010 would use the dynamic TOU map when generating TOU map data for the time between 10:00 AM and 2:00 PM on August 2, but would use its standard TOU map for the rest of the month.

TOU Maps in Detail

This section provides details concerning the TOU map objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom TOU map objects you create as part of your implementation. This section includes:

- A description of the D1-TOUMAP maintenance object
- Lists of the base package TOU map business objects, including “lite” business objects
- Details concerning TOU map-specific configuration options
- A sample TOU map business object (D2-TOUMap)

Maintenance Object - D1-TOUMAP

Service point business objects use the D1-TOUMAP maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-TOUMAP
Description	Time of Use Map
Service Name	D1-TOUMAP (Time of Use Maintenance)
Tables	<ul style="list-style-type: none"> • D1_TOU_MAP (Time of Use Map) - Primary • D1_TOU_MAP_CHAR (Time of Use Map Characteristics) - Child • D1_TOU_MAP_DYN_OPT (Time of Use Map Dynamic Option) - Child • D1_TOU_MAP_L (Time of Use Map Language) - Child • D1_TOU_MAP_LOG (Time of Use Map Log) - Child • D1_TOU_MAP_LOG_PARM (Time of Use Map Log Parameter) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Meter Data Management Base Package TOU Map Business Objects

The Oracle Utilities Meter Data Management base package includes the following TOU map business objects:

Business Object Name	Description
D2-TOUMap	Time of Use Map Instances of this business object represent individual time of use maps defined in the system.

The base package includes the following “lite” service point business objects:

Business Object Name	Description
D2-TOUMapLite	Time of Use Map LITE

Business Object Name	Description
D2-TOUMapParentLITE	Time of Use Map Parent LITE

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by TOU map business objects.

Business Object Options

TOU map business objects can make use of the following business object options:

- **TOU Map Data Business Object:** This option defines the business object used for TOU map data created by TOU maps created from this business object.

System Events

TOU map business objects can make use of the following system events:

- **TOU Map (BO) - Create TOU Map Data:** This system event defines the algorithm used for creating TOU map data for TOU maps created from this business object.

Example TOU Map - D2-TOUMap

The table below lists the details of the D2-TOUMap TOU map business object.

Option	Description
Business Object	D2-TOUMap
Description	Time of Use Map
Maintenance Object	D1-TOUMAP (Time of Use Map)
Application Service	D1-TOUMAPBOAS (TOU Map BO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> • TOU Map Data Business Object: D2-TOUMapData (TOU Map Data) • Display UI Map: D2-TOUMapDisplay (TOU Map - Display) • Portal Navigation Option: d1toumapTabMenu (TOU Map) • Display Map Service Script: D2-TOUMapDis (TOU Map Display) • Maintenance UI Map: D2-TOUMapMaint (TOU Map - Maintenance)
Algorithms	<ul style="list-style-type: none"> • TOU Map (BO) - Create TOU Map Data: D2-CRETMD-CT (Create TOU Map Data) • Pre-Processing: D2-DEL-TOUMD (Delete TOU Map Data) • Validation: D2-TOUMAP-VL (TOU Map Validation)
Lifecycle	<ul style="list-style-type: none"> • Active (Initial) • Inactive (Final)

Use the Business Object portal to view additional details concerning this business object.

Dynamic Options in Detail

This section provides details concerning the dynamic option objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom dynamic option objects you create as part of your implementation. This section includes:

- A description of the D1-DOP maintenance object
- Lists of the base package dynamic option business objects
- A sample dynamic option business object (D2-DynamicOption)

Maintenance Object - D1-DOP

Dynamic option business objects use the D1-DOP maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-DOP
Description	Dynamic Option
Service Name	D1-DOP (Dynamic Option Maintenance)
Tables	<ul style="list-style-type: none">• D1_DYN_OPT (Dynamic Option) - Primary• D1_DYN_OPT_CHAR (Dynamic Option Characteristics) - Child• D1_DYN_OPT_L (Dynamic Option Language) - Child• D1_DYN_OPT_LOG (Dynamic Option Log) - Child• D1_DYN_OPT_LOG_PARM (Dynamic Option Log Parameter) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Meter Data Management Base Package Dynamic Option Business Objects

The base package includes the following dynamic option business objects:

Business Object Name	Description
D2-DynamicOption	Dynamic Option Instances of this business object represent individual dynamic options defined in the system.

Example Dynamic Option - D2-DynamicOption

The table below lists the details of the D2-DynamicOption dynamic option business object.

Option	Description
Business Object	D2-DynamicOption
Description	Dynamic Option
Maintenance Object	D1-DOP (Dynamic Option)
Application Service	D1-DOPBOAS (Dynamic Option BO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> Related Transaction BO: D2-DynamicOptionEvent (Dynamic Option Event) Display UI Map: D2-DyOpDisplay (Dynamic Option - Display) Portal Navigation Option: d1dyopmTabMenu (Dynamic Option) Display Map Service Script: D2-DynOpSt (Dynamic Option Status Description) Maintenance UI Map: D2-DyOpMaint (Dynamic Option - Maintenance)
Algorithms	<ul style="list-style-type: none"> Information: D2-DOP-INFO (Dynamic Option Information)
Lifecycle	<ul style="list-style-type: none"> Active (Initial)

Use the Business Object portal to view additional details concerning this business object.

Dynamic Option Events in Detail

This section provides details concerning the dynamic option event objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom dynamic option event objects you create as part of your implementation. This section includes:

- A description of the D1-DOPEVT maintenance object
- Lists of the base package dynamic option event business objects
- A sample dynamic option event business object (D2-DynamicOptionEvent)

Maintenance Object - D1-DOPEVT

Dynamic option business objects use the D1-DOPEVT maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-DOPEVT
Description	Dynamic Option Event
Service Name	D1-DOPEVT (Dynamic Option Event Maintenance)
Tables	<ul style="list-style-type: none">• D1_DYN_OPT_EVENT (Dynamic Option Event) - Primary• D1_DYN_OPT_EVENT_CHAR (Dynamic Option Event Characteristics) - Child• D1_DYN_OPT_EVENT_LOG (Dynamic Option Event Log) - Child• D1_DYN_OPT_EVENT_LOG_PARM (Dynamic Option Event Log Parameter) - Child

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Meter Data Management Base Package Dynamic Option Event Business Objects

The Oracle Utilities Meter Data Management base package includes the following dynamic option business objects:

Business Object Name	Description
D2-DynamicOptionEvent	Dynamic Option Event Instances of this business object represent individual dynamic option events defined in the system.

Example Dynamic Option - D2-DynamicOptionEvent

The table below lists the details of the D2-DynamicOptionEvent dynamic option event business object.

Option	Description
Business Object	D2-DynamicOptionEvent
Description	Dynamic Option Event
Maintenance Object	D1-DOPEVT (Dynamic Option)
Application Service	D1-DOPEVTBOAS (Dynamic Option Event BO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> • Display UI Map: D2-DyOpEDisplay (Dynamic Option Event - Display) • Portal Navigation Option: d1dyopemTabMenu (Dynamic Option Event) • Pre-Processing Service Script: D2-DyOpEPre (Dynamic Option Event Preprocessing Service Script) • Display Map Service Script: D2-DyOpEStat (Dynamic Option Event Status Description) • Maintenance UI Map: D2-DyOpEMaint (Dynamic Option Event - Maintenance)
Algorithms	<ul style="list-style-type: none"> • Information: D2-DOPEVTINF (Dynamic Option Event Information) • Pre-Processing: D2-DYCOPEPR (Shift Dynamic Option Event Date/Times into Standard Time)
Lifecycle	<ul style="list-style-type: none"> • Frozen (Initial)

Use the Business Object portal to view additional details concerning this business object.

Configuring TOU Maps and Dynamic Options

This section provides high-level overviews of the steps involved in configuring custom TOU maps, dynamic options, and dynamic option events. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

Note: The procedures below focus on specific configuration tasks and options related to each of the objects described in this chapter, and do not address all the steps involved in creating business objects, UI maps, algorithms, etc. For more information about these subjects, refer to the Oracle Utilities Application Framework documentation.

Configuring Custom TOU Maps

Configuring custom TOU maps involves the following steps:

1. Design the TOU map business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom TOU map-related configuration objects required for your business objects, including:

Business Object Options: Create business objects for the following business object options:

- TOU Map Data Business Object

System Events: Create algorithms for the following system events:

- TOU Map (BO) - Create TOU Map Data

3. Create your TOU map business objects, referencing the configuration objects created above as appropriate.
4. Set up admin records that define the TOU map types you will use in your implementation.

Configuring Custom Dynamic Options

Configuring custom dynamic options involves the following steps:

1. Design the dynamic option business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom dynamic option-related configuration objects required for your business objects.
3. Create your dynamic option business objects, referencing the configuration objects created above as appropriate.

Configuring Custom Dynamic Option Events

Configuring custom dynamic option events involves the following steps:

1. Design the dynamic option event business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom dynamic option event-related configuration objects required for your business objects.
3. Create your dynamic option event business objects, referencing the configuration objects created above as appropriate.

Chapter 12

Usage Transactions

This chapter provides descriptions of usage transactions, including:

- **Understanding Usage Transactions**
- **Usage Transactions In Detail**
- **Configuring Usage Transactions**

Understanding Usage Transactions

This section describes usage transactions and their role in the usage calculation process.

Usage Transactions Overview

Usage transactions are records of bill determinant calculations for a usage subscription. Attributes used to define usage transactions can include the following:

- **Parent Usage Transaction:** The "parent" usage transaction for this "sub" usage transaction (if applicable). "Master" and "sub" usage transaction allow for hierarchical relationships between usage transactions. Applicable only to "sub" usage transactions.
- **Usage Subscription:** The usage subscription for which the usage transaction was created.
- **Previous Usage Transaction:** The previous usage transaction for the referenced usage subscription.
- **Usage External ID:** An ID used by external systems used to identify the usage transaction. When Oracle Utilities Meter Data Management is integrated with Oracle Utilities Customer Care and Billing, Oracle Utilities Customer Care and Billing can send this to help identify the previous usage transaction.
- **Status:** the current state of the usage transaction
- **Start and End Date and Times:** The start and end times that define the time period for the usage transaction.
- **Usage Group:** The usage group (or groups) used to calculate usage for the usage transaction.
- **Bill Condition:** The status of the bill associated with the usage transaction (if applicable), used for informational purposes only. Valid values include Initial, Interim, and Closing.
- **Automated Retry:** A flag that indicates if creation of the usage transaction should be automatically retried in the event of an error.
- **Next Retry Date/Time:** The date and time of the next attempt to create the usage transaction (applicable only if the **Automated Retry** flag is set to "Yes").
- **Retry Until Date/Time:** The date time until which the system will attempt to retry creating the usage transaction in the event of an error (applicable only if the **Automated Retry** flag is set to "Yes").
- **Sub Usage Transactions Exist:** A flag that indicates (Yes or No) if the usage transaction is a "parent" transaction and if sub-usage transaction exist for the usage transaction.
- **Defer Calculation:** A flag that indicates (Yes or No) if calculation of the usage transaction should be deferred.
- **Trace:** Indicates if tracing is on for the usage transaction.
- **Skip:** A flag that indicates if the usage transaction should be skipped.
- **Skip Reason:** A reason for skipping the usage transaction.
- **Next Scheduled Read Date:** The next scheduled read date for the device from which the measurements used to create the usage transaction came.
-
- **Calculation Mode:** A flag that indicates the mode in which the usage transaction calculations are/were performed. Valid values include:
 - **Hypothetical Calculation:** Indicates that the calculations are performed for purposes of comparing an existing rate against an alternative rate. This field can be referenced by usage rule eligibility criteria to determine if/when a usage rule should be processed when performing a rate comparison.

- **Usage Adjustment Profile Factors:** Defines one or more profile factors and applicable characteristic values used to adjust usage values when performing a rate comparison.

Usage adjustment profile factors are defined for specific rates in the Rate Comparison Configuration section of the "Self-Service Master Configuration". When a configured rate is used in usage transaction calculations, the profile factors defined for that rate are retrieved by "Profile Accumulation" usage rules.
- **Interval Measuring Component:** Start and End dates and times for the interval measuring component that created the measurements used in calculating usage for the usage transaction (if applicable).
- **Scalar Measuring Component:** Start and End dates and times (and other details) for the scalar measuring component that created the measurements used in calculating usage for the usage transaction (if applicable).
- **Date Break:** One or more date breaks for the usage transaction. Date breaks are used to break up a usage period into sub-periods based on the dates on which rate changes took place for the service point (and its related account).
- **Issues:** A list of issues or exceptions related to the usage transaction
- **Scalar Detail:** Details of scalar measurements used in calculating for the usage transaction
- **Usage Period:** A list of the service quantities calculated for the usage transaction, including the following:
 - **UOM, TOU, SQI** (as applicable): The unit of measure/time of use period/service quantity identifier for the service quantity
 - **Quantity:** The total quantity for the service quantity
 - **Data Quality Assessment:** Indicates the quality of the service quantity. Can be one of the following:
 - **No Assessment Done:** Indicates that quality assessment was not performed for the service quantity
 - **Quantity Based on Regular Data Only:** Indicates that the service quantity contains only data with a condition that falls in the "regular" condition range (defined by the Regular Top Range Condition and Regular Bottom Range Condition parameters in the "Apply Usage Rule" algorithm used to calculate the service quantity)
 - **Quantity Based on Some Non-Regular Data:** Indicates that the service quantity contains some data with a "non-regular" condition code (i.e. data that falls outside the "regular" condition range)
 - **SQ (Service Quantity) Type:** The type of service quantity (Factor, Measuring Component, Other)
 - **Service Point:** The service point associated with the service quantity
 - **Measuring Component:** The measuring component associated with the measurements used to calculate the service quantity
 - **TOU Map** (if applicable): The time of use map used to calculate the service quantity
 - **Usage Group:** The usage group that contains the usage rule used to calculate the service quantity
 - **Usage Rule:** The usage rule used to calculate the service quantity
 - **Source Measurement Quality List:** XML that contains a list of measurements aggregated by condition/quality.

- **SQ Highlight Date/Time List:** XML that contains a list of maximum/minimum date/times, based on condition/quality.
- **Item Detail:** A list of one or more items for which usage was calculated for the usage transaction. This section displays details for each service quantity calculated for each item, including item type, item count, total quantity, daily service quantity, start and end times, UOM, service point, override quantity (if applicable) and the usage group/rule used to calculate the usage.

Usage Transactions - A Closer Look

This section provides more details concerning some specific aspects of usage transactions.

Calculation Period

Usage transaction requests must specify the date range for the usage transaction. This date range is referred to as the calculation period for the usage transaction. The dates that define the calculation period are specified by a subscribing system when it requests a usage transaction.

Multiple Device Configurations and the Calculation Period

In situations where a meter exchange has taken place at a service point, more than one device configuration can be in effect during the usage transactions cutoff period (the time period between the **End Date/Time From** and **End Date/Time To** values).

To account for this, the **Meter Exchange Option** on the usage subscription type can be used to specify how the usage calculation should handle multiple device configurations in effect during the cutoff period. If the flag is set to 'Defer Calculation', new device configurations are excluded in the current usage transaction. If the flag is blank or set to 'Calculate Usage', new device configurations are included in the current usage transaction.

Date Breaks

As noted above, date breaks are used to break up a usage period into sub-periods based on the dates on which rate changes took place for the service point. For example, suppose a subscribing system requests usage for the month of January. The customer for this request has an interval meter, and the customer's usage is calculated by applying a TOU map to their interval consumption. The subscribing system detects that the customer's rate changed in the middle of January (on January 16) and wants the TOU consumption calculated in two "chunks" (before and after the rate change). Because the customer has an interval meter, the exact consumption amounts before and after the rate change can be precisely calculated (as opposed to calculating each period's amount by dividing the total usage by the number of days in each period).

One approach to this situation would be for the subscribing system to request two usage transactions (where each has the desired date range). Another approach is for the subscribing system to request a single usage transaction with date breaks that define the date ranges before and after the rate change. To continue the above example, a usage transaction could be created with a date break on January 16.

If a usage transaction has date break(s), the usage calculation engine segregates the usage into multiple usage periods based on the date breaks. If there are no date breaks in the usage transaction, a single usage period is created for the entire calculation period.

As of the v2.0.0 release, the only way a usage transaction can have date breaks is if the subscribing system supplies these when it requests usage (only the subscribing system knows if and when its prices and pricing rules change during the billing period). If an implementation has additional criteria that causes date breaks, these criteria can be easily added to the usage transaction's business rules.

Service Quantities

Every usage period created for a usage transaction contains one or more service quantities.

Service quantities are calculated by the usage group's rules specified for the usage subscription from which the usage transaction is created. For example, a usage transaction might have service quantities calculated by a single rule that applies a TOU map to the kWh channel on the device configuration installed at the service point.

Each service quantity lists details about that quantity, including a UOM (or TOU or SQI as appropriate) and a quantity. In addition, each service quantity also references the source (measuring component) of the quantities (for audit purposes).

The base-package usage rules can calculate different types of service quantities. For example, a usage group with two rules might create the following types of service quantities:

- One that retrieves and snapshots the interval for the kWh channel
- One that applies a TOU map to the channel and saves the results

Performance Note: the base-package rule that snapshots intervals can create a large amount of data. This should only be used if the subscribing system requires the intervals. For example, you should not snapshot intervals on a usage transaction for audit purposes.

Quality Assessment and Bill Print Options

Usage transaction service quantities can also contain information about the quality of the measurements used in their calculations and other information that can be sent to an external customer information or bill print system. This information is included in both the usage transaction and its related outbound message.

Quality Assessment

The “Get Interval Data”, “Get Scalar Details”, and “Get Time of Use Mapped Usage” usage rules include a “Perform Measurement Quality Assessment” option that indicates (yes or no) if the measurement conditions of the measurements used when calculating usage should be evaluated and included in the resulting usage transaction and outbound message. When enabled (set to “Yes”), the usage rule algorithm evaluates the condition codes of the measurements used in the calculation to determine if they fall within a “Regular” range of conditions (this range is defined by the “Bottom Regular Condition Range” and “Top Regular Condition Range” algorithm parameters). The result of this assessment is displayed in the “Data Quality Assessment” column in the Usage Periods section which indicates if the service quantity is based on “regular” or “non-regular” data. (If the “Perform Measurement Quality Assessment” option is disabled, the “Data Quality Assessment” column displays “No Assessment Done”). In addition, the “Source Measurement Quality List” column contains XML that contains a list of measurements aggregated by condition/quality.

Maximum/Minimum Dates and Times

The “Get Interval Data”, “Get Time of Use Mapped Usage”, and “Vector and Service Quantity Math” usage rules can be configured to identify the date/time of the maximum or minimum measurement value used when calculating the service quantity. For these functions, if “Max” or “Min” is selected as the “Calculation Function” (or the “Comax” and “Comin” functions for the “Vector and Service Quantity Math” rule), the date/time of the maximum/minimum value will be included in the resulting usage transaction and outbound message. The “SQ Highlight Date/Time List” column contains XML that contains a list of maximum/minimum date/times, based on condition/quality.

Usage Transaction Export Configuration

Usage transactions can also include additional information about its related meters and service points, as well as an indication if the service quantity is based on estimated usage. The “Usage Transaction Export Configuration” master configuration is used to control the export of this data. It holds the device and service point business objects used to supply meter and service point information included in the usage transaction and outbound message. This information includes the following:

- Device identifier information within the “Summary Usage Periods” section of the outbound message includes:
 - Badge Number
 - Serial Number
 - From Date/Time
 - To Date/Time

- Device identifier information with the “Scalar Details” section of the outbound message includes:
 - Badge Number
 - Serial Number
- Additional service point information within the “Summary Usage Periods” section includes:
 - External SP ID
 - External Premise ID
 - Market

The “Usage Transaction Export Configuration” master configuration also defines the threshold percentage above which a service quantity being exported via the usage transaction outbound message is marked as an estimate. For example, if the “Estimation Threshold Percentage” is set to 35%, any service quantity being exported that was calculated from measurements containing more than 35% estimated usage is marked “estimated”.

Service Quantities for Scalar Usage

Usage transactions and service quantities for scalar usage differ in some ways from those created from interval usage. For example, suppose a subscribing system requests usage for the month of January for a customer with a scalar meter that has been exchanged mid-month. In this case, the customer's usage is calculated by finding the scalar readings in the requested period (including all meter exchanges), and the subscribing system requires a record of all scalar readings AND a total of their consumption.

Scalar rules cannot use date breaks because the system doesn't store interval values so it cannot accurately compute the amount in each period. In this case, scalar usage rules retrieve scalar readings for the service points linked to the usage subscription, and then creates a single usage period for the usage transaction's entire calculation period. The individual scalar readings are captured as “scalar details” in the usage transaction.

To continue the example above, there would be 2 entries in the Scalar Details, one for each meter, and a single usage period that contains the total consumption for the entire calculation period. The tables below illustrate what this might look like.

Scalar Details:

Final Usage	Usage Group	Measured Usage UOM/TOU/SQI	Service Point	Measuring Component	Start Reading	Stop Reading
202 BTU	Residential / Retrieve Scalar	200 CCF	39191912312 (18 Main St)	8382821921 (CCF register)	1-Jan-10 3pm 10000	15-Jan-10 6pm 10200
303 BTU	Residential / Retrieve Scalar	300 CCF	39191912312 (18 Main St)	8382821921 (CCF register)	15-Jan-10 6pm 10200	30-Jan-10 4pm 10500

Usage Period:

Date/Time Range	
1-Jan-10 3:00pm to 30-Jan-10 4:00pm	
UOM/TOU/SQ	Usage
BTU	505 BTU

This example also illustrates how a usage rule can convert a Measured UOM (CCF) into a Final UOM (therms).

Item Details

When calculating usage for service points at which items are installed, item details are also created in the usage transaction. Item details include details for each service quantity calculated for each item, including item type, item count, total quantity, daily service quantity start and end times, UOM, service point, override quantity (if applicable) and the usage group/rule used to calculate the usage.

Calculating Usage for Items

Usage transactions can also calculate usage for items installed at service points. Usage for items is calculated through use of “Get Item Counts and Consumption” usage rules. These rules calculate usage based on the number of items installed at each service point and the average daily service quantity defined for each item's type. The specifics of how usage is calculated for items is based on the service point category (defined on the service point type) at which the items are installed (item or multi-item).

See **Get Item Counts and Consumption** in the **Base Package Usage Rules** section of the *Oracle Utilities Meter Data Management User's Guide* for more information about these usage rules.

Hypothetical Calculations - Rate Comparisons

When Oracle Utilities Meter Data Management is integrated with Oracle Utilities Customer Self Service, customers can perform rate comparison calculations to see the impact to their usage and billing charges based on a set of user-defined usage adjustments. Rate comparison usage transactions are flagged “Hypothetical Calculation” in the **Calculation Mode** field.

Profile Accumulation Usage Rules and Usage Adjustment Profile Factors

Rate comparison usage transactions can be created through the use of “Profile Accumulation” usage rules. These rules combine a service point's measurements with “usage adjustment” measurements derived from measurement data for profile measuring components defined as characteristic values for one or more profile factors.

Usage adjustment profile factors are associated to usage adjustment types defined in the Usage Adjustment Type extendable lookup and rates defined in the CC&B Rate Schedule extendable lookup (this association is defined in the Self-Service Master Configuration). When the usage transaction is created, the rate determines the usage group to execute and usage adjustment profile factors to include in the calculation based on the CC&B Rate Schedule extendable lookup.

Note: Profile accumulation usage rules should be configured with eligibility criteria such that they are only executed when the Calculation Mode for the usage transaction is set to “Hypothetical Calculation (D2HC)”. See **Profile Accumulation** in the **Base Package Usage Rules** section of the *Oracle Utilities Meter Data Management User's Guide* for more information about these usage rules.

For More Information

Refer to *Oracle Utilities Customer Self-Service Implementation Guide* and the *Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 3.1.1 Implementation Guide* for more information about configuring Oracle Utilities Meter Data Management to support rate comparison calculations.

Usage Transaction Corrections Based on Initial Measurement

In some situations, existing usage transactions can be impacted by incoming initial measurement data if the start and stop date/time of the initial measurement overlaps with the date range of the usage transaction. The “Finalized” state of the initial measurement business objects include algorithm (D1-TRAN-UT) that creates a “Usage Transaction Correction Processor” activity for each usage transaction that may have been impacted by an initial measurement.

This algorithm retrieves all usage transactions that overlap with the current initial measurement's Start and End Date/Times and that are linked to a usage subscription associated to a service point at which the device is installed for the initial measurement's measuring component. The usage

transaction selection filters out the sub usage transactions and processes only “parent” usage transactions.

For each usage transaction retrieved, the algorithm checks to see whether there is a Usage Transaction Correction Processor activity (of the same business object as referenced in the algorithm’s soft parameter) in a non-final state that references the current usage transaction. If so, the current usage transaction is ignored and the next usage transaction is processed among those impacted by the current initial measurement.

If there is no Usage Transaction Correction Processor activity existing for the usage transaction (or if the only Usage Transaction Correction Processors existing for the usage transaction are in a final state), a new Usage Transaction Correction Processor activity is created using the business object referenced in the soft parameter (If the soft parameter is left blank or an active Activity Type cannot be found for the business object referenced in the soft parameter, no Usage Transaction Correction Processor activity will be created).

Note: Usage Transaction Correction Processor activities are NOT created if/when final measurements are not updated as a result of an incoming initial measurement.

Usage Transaction Exceptions

Usage rules can both calculate usage and validate that the calculated usage is reasonable. The usage calculation process creates an exception for each problem encountered. Multiple exceptions can be created when a usage transaction is subject to validation. This allows users to see all of the problems detected by the usage calculate process.

Note: Usage exceptions are held within the usage transaction (in its CLOB). This contrast to VEE exceptions is intentional as usage transaction exceptions should be very rare whereas VEE exceptions are relatively common.

Exception Categories

Similar to VEE exceptions, there are three categories, or severities of usage exceptions:

- **Info:** Used to highlight something interesting, but not sufficient to cause the usage calculation to be put into the exception state.
- **Issue:** Used to report a problem that will prevent the usage transaction from being finalized. Multiple "issue exceptions" can be created during usage calculation. If at least one issue exists after all rules have been applied, the usage transaction is transitioned to the exception state.
- **Terminate:** Used to report a severe issue that will cause the usage calculation process to stop and the usage transaction to be transitioned immediately to the exception state. Only one terminate exception can be issued (as the first one causes usage calculation to stop).

Exceptions and To Do Entries

In addition to exceptions, usage processing can also trigger the creation of To Do Entries related to failed validations.

If Issue or terminate exceptions exist for an initial measurement, a To Do Entry is created when the usage transaction is transitioned to the Exception state. The To Do Type and default To Do Role of this To Do Entry are defined on the Enter system event for the Exception state of the business object used to define the usage transaction.

To Do Entries created in this way can be routed to different roles depending on the exception's message category and number (using the To Do Type's Message Overrides tab).

Available Actions for Usage Transactions with Exceptions

Users have a number of options for dealing with usage transactions with exceptions.

- After correcting the cause of the issues that triggered the exceptions, a user can re-calculate the usage transaction.
- A user can discard the usage transaction.

- A user can manually complete the usage transaction. This sends the usage transaction to the subscribing system “as is”
- A user can edit the Post VEE quantities (if necessary) and manually complete the initial measurement. This will cause final measurements to be created using the contents of the Post VEE quantities.

Note: No VEE processing is performed on manually completed initial measurement data.

Regardless of the action taken by the user, the system will complete any open To Do Entries that created when the usage transaction entered the Exception state.

Creating Usage Transactions from External Systems

When Oracle Utilities Meter Data Management is integrated with a customer information system (such as Oracle Utilities Customer Care and Billing), usage transactions can be created via a request from the customer information system.

To invoke a usage transaction request, the external system must invoke an XAI Inbound Service mapped to the Usage Transaction Seeder (D2-UsgTranSeeder) business object. This business object does the following:

- Determines the usage subscription ID based on an external usage subscription ID

This processing is performed via the Determine Usage Subscription ID (D2-DETUSID) Pre-Processing algorithm.

- Determines the appropriate usage transaction business object to create.

This processing is performed via the Determine Usage Transactions Business Object (D2-DETUTBO) Pre-Processing algorithm.

This algorithm uses the “How To Create Usage Subscription Related Information” (D2-HowToCreateUSInformation) processing method defined for the “Usage Transaction Creation” processing role on the service provider that represents the external system.

In addition to the Usage Transaction Seeder business object, the Oracle Utilities Meter Data Management base package also includes the following XAI configuration options to support creating usage transactions from external systems:

- **XAI Inbound Service:** D2-UsageTransactionRequestInbound (D264745327)

This inbound service is mapped to the Usage Transaction Seeder (D2-UsgTranSeeder) business object.

Sending Usage Transactions to External Systems

When Oracle Utilities Meter Data Management is integrated with a customer information system (such as Oracle Utilities Customer Care and Billing), usage transaction information, including bill determinants and other data can be sent to the customer information system.

Usage transactions are sent to subscribing systems when they enter the “Sent” state. The “Send Usage” (D2-SEND-USG) Enter algorithm on the Sent state of the base package usage transaction business object (D2-UsageTransaction) determines the method used to send the information, based on the Usage Recipient (service provider) defined on the usage transaction’s Usage Subscription. Usage transactions can be sent to service providers via either online real-time processing or periodically via batch processing.

Online Real-Time Processing

To set up the service provider to support online real-time notification of usage transactions, do the following:

- Create one or more Outbound Message Types that reference the outbound message business object to be used to send usage transaction information to the external system. The base package include the following business object for this:
 - Usage Transaction Outbound Message (D2-UsageTranOutboundMesg)
- Define an XAI Sender that will be used to send the message to the external system.
- Add the outbound message type to the service provider's External System and reference the XAI sender created above.
- Add a processing method to the service provider as follows:
 - **Processing Role:** Usage Transaction Notification - Online
 - **Processing Method:** How To Send US Related Information
 - **Status:** Active
 - **Default Processing Method:**
 - **Outbound Message Type:** the outbound message type created above
 - **Override Processing Method:** outbound message types for specific usage subscription types if applicable

Batch Processing

To set up the service provider to support periodic batch processing of usage transactions, do the following:

- Create one or more batch controls that will send the usage transaction. These batch programs should invoke the business objects that will contain the usage transaction information. The base package include the following business object for this:
 - Usage Transaction Outbound Message (D2-UsageTranOutboundMesg)
- Add a processing method to the service provider as follows:
 - **Processing Role:** Usage Transaction Notification - Batch
 - **Processing Method:** How To Send US Related Information
 - **Status:** Active
 - **Default Processing Method:**
 - **Batch Control:** the batch control created above
 - **Override Processing Method:** batch controls for specific usage subscription types if applicable

Including Summary Service Quantities and Interval Data Snapshots with Usage Transactions

Usage transactions can be configured to include summary service quantities and an interval data "snapshot" along with service quantities and other information sent to external systems. This is used when sending usage to a billing system that performs interval-based pricing calculations.

Only usage transactions created (in whole or in part) by either the "Get Interval Data" or "Vector and Service Quantity Math" usage rules can include interval data snapshot information. These rules use the **Extract Interval Data** flag to indicate ("Yes" or "No") if a "snapshot" of the interval data resulting from the usage rule should be sent to external systems when usage from the usage transaction is sent to external systems.

In the base package, the Usage Transaction Outbound Message (D2-UsageTranOutboundMesg) business object is used to create outbound messages containing usage transactions. This business

object includes the “Build Summary SQs” (D2-BLDSUMSQ) Pre-Processing algorithm which calculates and creates the interval data snapshot and other summary service quantities.

How Summary Service Quantities and the Interval Data Snapshot is Calculated

As noted above, the “Build Summary SQs” (D2-BLDSUMSQ) algorithm builds the Summary Usage Periods List based on the usage transaction’s details. For every usage period within the usage transaction:

- An entry is inserted into the summary service quantity list for every distinct UOM/TOU/SQI combination in the usage period service quantity list.
- An entry is inserted into the service point service quantity list for every distinct service point ID/UOM/TOU/SQI combination in the usage period service quantity list.

Interval data for multiple service quantities are combined into a single interval data curve per UOM/TOU/SQI combination. Intervals are summarized and scaled based on a common SPI

The “Build Summary SQs” (D2-BLDSUMSQ) algorithm uses the “Target Interval Size” parameter to define the interval size of the combined curve when one or more curves are combined with different interval sizes. For example, when combining interval data with 15 minutes intervals (SPI 900) and data with 60 minute intervals (SPI 3600), a Target Interval Size of 60 minutes (SPI of 3600) would result in combined data with an interval size of 3600 (the 15 minute intervals would be scaled up to 60 minute intervals before being combined with the 60 minute interval data).

Note: The condition code of combined interval values is always set to “DERIVED” (999999)

Note: When summarizing data, missing intervals are replaced with "0" value intervals with a condition code of "No Read - System" (100000).

Persisting Outbound Messages - For Testing ONLY

By default, the “Send Usage” (D2-SEND-USG) algorithm does not persist outbound messages. When testing your implementation, it is useful to change this behavior so that outbound messages sent containing usage transactions are persisted in the system.

To enable persistence of outbound messages created by the “Send Usage” (D2-SEND-USG) algorithm, edit the following lines (underlined) in the “Send Usage” (D2-SendUsage) plug-in script:

Base Package:

```
// Invoke Outbound Message Dispatcher Business Service
move 'false' to "F1-OutmsgDispatcher/persist";
move 'true' to "F1-OutmsgDispatcher/trapError";
move "F1-OutboundMessageType/bo" to "F1-OutmsgDispatcher/businessObjectName";
move "utOutboundMsgBo_schema" to "F1-OutmsgDispatcher/businessObjectRequest";
invokeBS 'F1-OutmsgDispatcher' using "F1-OutmsgDispatcher";
```

Edited:

```
// Invoke Outbound Message Dispatcher Business Service
move 'true' to "F1-OutmsgDispatcher/persist";
move 'true' to "F1-OutmsgDispatcher/trapError";
move "F1-OutboundMessageType/bo" to "F1-OutmsgDispatcher/businessObjectName";
move "utOutboundMsgBo_schema" to "F1-OutmsgDispatcher/businessObjectRequest";
invokeBS 'F1-OutmsgDispatcher' using "F1-OutmsgDispatcher";
```

Usage Transactions In Detail

This section provides details concerning the usage transaction objects supplied as part of the base package. This information illustrates how the base package objects were designed, and can serve as the basis for any custom usage transaction objects you create as part of your implementation. This section includes:

- A description of the D1-USAGETRANS maintenance object
- Lists of the base package usage transaction business objects, including “lite” business objects
- A sample usage transaction business object (D2-UsageTransaction)

Maintenance Object - D1-USAGETRAN

Usage transaction business objects use the D1-US maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-USAGETRAN
Description	Usage Transaction
Service Name	D1-USAGETRAN (Usage Transaction Maintenance)
Tables	<ul style="list-style-type: none"> • D1_USAGE (Usage) - Primary • D1_USAGE_CHAR (Usage Characteristics) - Child • D1_USAGE_LOG (Usage Log) - Child • D1_USAGE_LOG_PARM (Usage Log Parameters) - Child • D1_USAGE_PERIOD (Usage Period) - Child • D1_USAGE_PERIOD_ITEM_DET (Usage Period Item Detail) - Child • D1_USAGE_PERIOD_SQ (Usage Period Service Quantity) - Child • D1_USAGE_REL (Usage Relationship) - Child • D1_USAGE_SCALAR_DTL (Usage Scalar Detail) - Child
Algorithms	<ul style="list-style-type: none"> • Information: D1-USAGEINFO (Usage Transaction Information)

Use the Maintenance Object portal and the Application Viewer to view more details about this maintenance object.

Meter Data Management Base Package Usage Transaction Business Objects

The Oracle Utilities Meter Data Management base package includes the following usage transaction business objects:

Business Object Name	Description
D2-SubUsageTransaction	Sub Usage Transaction Instances of this business object represent individual sub usage transactions created in the system.
D2-UsageTransaction	Usage Transaction Instances of this business object represent individual usage transactions created in the system.

The base package includes the following “lite” usage transaction business objects:

Business Object Name	Description
D1-UsageTransactionParentLite	Usage Transaction Parent LITE
D2-UsageIntervalDataLite	Usage Transaction Interval Data LITE
D2-UsageTranExceptionsLITE	Usage Transaction Exceptions LITE
D2-UsageTranScalarDetailsLite	Usage Transaction Scalar Details LITE
D2-UsageTransactionDateBreaks	Usage Transaction Date Breaks LITE
D2-UsageTransactionExceptions	Usage Transaction Exceptions LITE
D2-UsageTransactionMini1	Usage Group Transaction Processing LITE
D2-UsageTransactionParentLITE	Usage Transaction Parent LITE
D2-UsageTransactionStatusLite	Usage Transaction Status LITE
D2-UsageTransacitonTrace	Usage Transaction Trace LITE
D2-UsageTransactionUsagePeriod	Usage Transaction Usage Periods LITE
D2-UsgTranChkSubLITE	Usage Transaction Check Sub LITE
D2-UsgTranDeferLITE	Usage Transaction Defer LITE
D2-UsgTranRetryLITE	Usage Transaction Retry LITE
D2-UsgTranSkipDetailsLite	Usage Transaction Skip Details LITE
D2-UsgTranSkipDetailsParentLITE	Usage Transaction Skip Details Parent LITE
D2-UsgTranSubLITE	Usage Transaction Sub LITE

The base package includes the following additional usage transaction business objects:

Business Object Name	Description
D2-UsgTranSeeder	Usage Transaction Seeder (used to determine the usage transaction business object to use when creating new usage transactions)

Example Usage Transaction - D2-UsageTransaction

The table below lists the details of the D2-UsageSubscription usage transaction business object.

Option/Field	Description
Business Object	D2-UsageTransaction
Description	Usage Transaction
Maintenance Object	D1-USAGETRAN (Usage transaction)
Application Service	D2-USAGETRANBOAS (Usage transaction BO)
Instance Control	Allow New Instances
Options	<ul style="list-style-type: none"> Display UI Map: D2-UsageTranDisplay (Usage Transaction - Display) Portal Navigation Option: d1utnmTabMenu (Usage Transaction) Pre-Processing Service Script: D2-UsgTraPre (Usage Transaction Pre-Processing) Display Map Service Script: D2-UsgRetDts (Usage Retrieve Details for Display) Maintenance UI Map: D2-UsageTranMaint (Usage Transaction - Maintenance)
Algorithms	<ul style="list-style-type: none"> Validation: D2-DEL-ACT (Delete Validation) Validation: D2-VALPREUT (Validate Previous Usage Transaction) Validation: D2-SQTYP-VAL (Validate Usage Transaction Service Quantity Entries) Validation: D2-INSCP-VAL (Interval and Scalar Period Validation)
Lifecycle	<ul style="list-style-type: none"> Pending (Initial) Calculation Deferred (Interim) Calculate (Interim) Calculation in Progress (Interim) Approval In Progress (Interim) Issue Detected (Interim) Approve (Interim) Sent (Final) Subsequent Correction (Final) Discarded (Final) Send Error (Interim)

Use the Business Object portal to view additional details concerning this business object.

Configuring Usage Transactions

This section provides high-level overviews of the steps involved in configuring custom usage transactions. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

Note: The procedures below focus on specific configuration tasks and options related to each of the objects described in this chapter, and do not address all the steps involved in creating business objects, UI maps, algorithms, etc. For more information about these subjects, refer to the Oracle Utilities Application Framework documentation.

Configuring Custom Usage Transactions

Configuring custom usage transactions involves the following steps:

1. Design the usage transaction business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom usage transaction-related configuration objects required for your business objects.
3. Create your usage transaction business objects, referencing the configuration objects created above as appropriate.

Chapter 13

Aggregations

This chapter describes aggregations and how they are performed in Oracle Utilities Meter Data Management, including:

- **Understanding Aggregations**
- **Configuring Aggregations**

Understanding Aggregations

This section describes aggregations and how they are created and calculated in Oracle Utilities Meter Data Management.

The term aggregation refers to the summarization of measurements and statistical values from a group of related measuring components or items. Aggregated values are used primarily for analysis. For example, an electric utility may want to see aggregated consumption for each transformer within its service territory, where a transformer is uniquely identified by 3 values:

- The Transformer itself,
- The Feeder associated with the Transformer, and
- The Substation associated with the Feeder.

Aggregator Measuring Components and Dimensions

Aggregator measuring components hold summarized usage from other measuring components or items. For example, aggregator measuring components could be configured to hold total consumption for each postal code within a service territory, or for each transformer within its service territory (as above). The aggregated values of an aggregator measuring component's constituent measuring components/items are its final measurements. Aggregator measuring components do NOT have initial measurements.

Every type of aggregation has one or more dimensions that define how to gather the constituent measuring components/items from which measurements are aggregated. For example, the transformer aggregation referenced above has 3 dimensions: Substation / Feeder / Transformer.

An aggregator measuring component must exist for every distinct combination of dimensions. For example, the transformer aggregation requires a separate measuring component for every distinct combination of substation / feeder / transformer found on the electric service points.

Any combination of dimensions is possible. The prior example illustrated an aggregator measuring component whose dimensions are all derived from the same type of object, in this case, the substation, feeder, and transformer defined on electric service points.

Aggregator measuring components can exist for dimensions derived from any data (including administrative data). For example:

- A Postal Code / Measuring Component Type aggregation has a separate aggregator measuring component for every combination of postal code and measuring component type
 - Postal code is an attribute of each service point
 - Measuring Component Type is an attribute of each measuring component
- A Rate / Postal Code aggregation has a separate aggregator measuring component for every combination of rate and postal code
 - Rate is an attribute of each usage subscription
 - Postal code is an attribute of each service point

The base package and demonstration data include several examples of different dimensional combinations. Implementations can introduce others based on their requirements.

Types of Aggregated Data

Any of the quantities on final measurements can be aggregated.

Interval measuring components of different magnitudes, such as KW vs MW, SPIs (15 minute vs 60 minute) and, in some cases, units of measure can be aggregated together.

Note: Not all units of measure can be aggregated together. Only those that share a common Base Unit of Measure can be aggregated together. For example, KW can be converted to kWh and vice versa, but BTU measurements cannot be converted to KW.

This requires each constituent measuring component to be converted into a common UOM and interval size. Converting different UOMs to a common UOM uses the Base Unit of Measure and Magnitude attributes. Converting differing interval sizes to a common interval size involves scaling the interval values up or down accordingly.

Both scalar and interval measuring components can also be aggregated together. This requires the scalar consumption to be intervalized into the aggregator measuring component's UOM and interval size prior to aggregation.

Aggregating Aggregations

Aggregations can also be aggregated into higher levels. For example, the aggregator measuring components whose measurements hold aggregated consumption for each Substation / Feeder / Transformer combination can be used to aggregate consumption at each Substation / Feeder.

In this case, distinct aggregator measuring components must exist for every Substation / Feeder combination. The measurements of these aggregator measuring components are the sum of the consumption on the transformers linked to each feeder.

Note that these values can be derived from the Substation / Feeder / Transformer aggregator measuring components, meaning that it is only necessary to aggregate the transformer measurements once.

In turn, aggregator measuring components that hold aggregated consumption for each Substation / Feeder combination could be used to aggregate consumption at each Substation. Again, distinct aggregator measuring components exist for every Substation, and their measurements would represent the sum of their feeders.

Finally, total consumption for all substations could be aggregated using the same technique.

Aggregation Calculations

The system periodically aggregates consumption via batch process, using a deferred monitor on the aggregator measuring components. In addition, users can re-aggregate data in real-time if they don't wish to wait for the batch process, or if the original aggregation needs to be re-calculated due to incorrect data. Users can also create ad hoc aggregations "on the fly" and these will persist in the database in the same manner as other aggregations.

Understanding Aggregation Periods

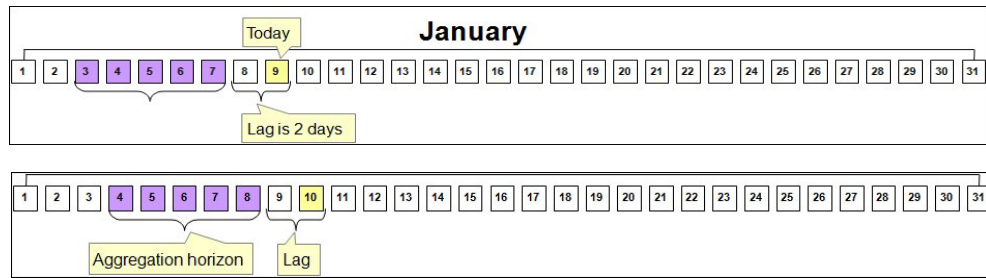
The start and end dates and times for aggregation calculations are based on the following:

- Aggregation Horizon
- Aggregation Lag
- Aggregation Cut Off Time

Whenever aggregation is performed for an aggregator measuring component, consumption is aggregated for every day in its "Aggregation Horizon." The "Aggregation Horizon" is the number of days during which there's a potential change in measurement data for one or more of the measuring components associated an aggregator measuring component.

Aggregation calculations typically lag behind the current date by a few days to give the system time to upload and perform validations and create final measurements. The amount of lag time is referred to as the "Aggregation Lag" and is the number of days between the date on which aggregation calculations are performed and the end date of the aggregation period. This defines the time period between the aggregation calculation date and the end of the aggregation horizon

that serves to allow all measurements to arrive. This together with the Aggregation Horizon is used to determine the start and end dates of an aggregation period. For example, with an Aggregation Horizon of 5 and an Aggregation Lag of 2, aggregation calculations performed on January 9 would be for an aggregation period of January 3 through January 7. The next day (January 10), the aggregation period would shift to January 4 through January 8.



Aggregation is always performed through a given "through time" (such as 12:00 AM) rather than through the actual time of the aggregation calculation. This time is referred to as the "Aggregation Cut Off Time." For example, the stop time for aggregation calculations with an Aggregation Lag of 2, and an Aggregation Cut Off Time of 10:00 PM will always be 10:00 PM 2 days prior to the date on which the calculations are performed. In the above examples, aggregations performed on January 9 would have an end date/time of 10:00 PM on January 7, and aggregations performed on January 10 would have an end date/time of 10:00 PM on January 8.

The Aggregation Horizon, Aggregation Lag, and Aggregation Cut Off Time are configured for each aggregator measuring component type.

Aggregation and Re-Aggregation

The use of the aggregation horizon means that aggregated totals for some days will be re-aggregated until those days are no longer covered by the aggregation horizon. For example, with an aggregation horizon of 5 days and an aggregation lag of 2 days, on the night of January 9 the aggregation period would be January 3 through January 7 (as in the above example).

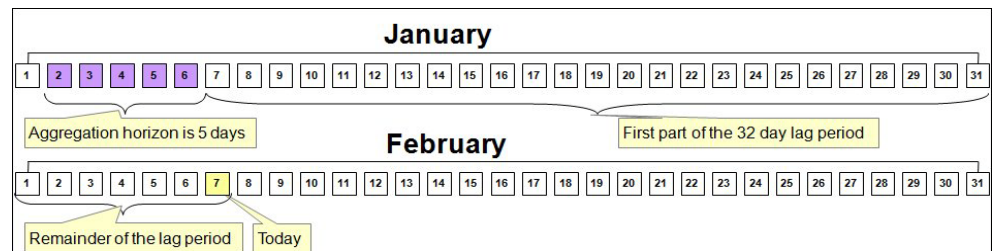
On the night of January 10, the horizon will shift 1 day (to January 4 through January 10, again as above). This means the following:

- The totals for January 3 calculated on January 9 will be untouched (January 3 now falls outside the aggregation horizon)
- The totals for January 4 through January 7 will be re-derived because corrections may have occurred (and they still fall within the aggregation horizon)
- The totals for January 8 will be calculated for the first time (because it now is within the aggregation horizon).

Manually Read Meters and Aggregation Lag

Aggregations that include manually read meters often have much longer aggregation lags than those for automatically read meters. This allows more time for manual meter readings to be imported into the system for use in aggregations.

For example, suppose a situation where manual meter reads arrive approximately 1 month after the date of the reading. In this case, it wouldn't be until a meter read upload on February 7 that the last manual reads including consumption from January 6 will exist. Since we don't want to perform aggregations for January 6 until there's a decent chance that all consumption for that date exists, an aggregation lag of 32 days ensures that the data for January 6 is in the system when the aggregation is performed on February 7.



When An Aggregator Measuring Component Contains Both Manual and AMI Measuring Components

The previous section suggests that if an aggregator measuring component contains both manually read and AMI measuring components, the aggregator measuring component should have a long lag time rather than the short time shown in the earlier example.

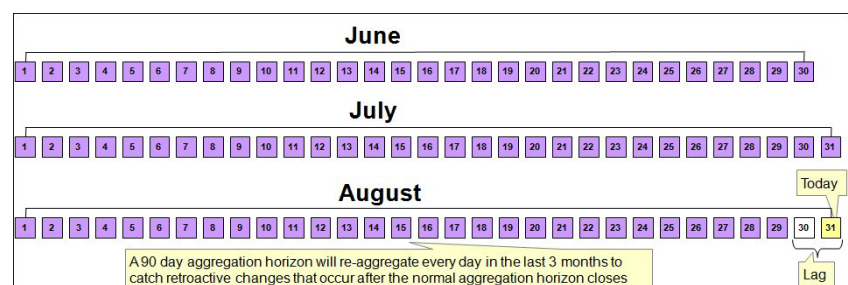
While this is one possible approach, an alternative approach could allow users to see timely aggregations of the AMI channels and only lag the manual channels. To do this they could configure the system as follows:

- Create an aggregator measuring component that only aggregates manual measuring component types (horizon: 5 days, lag: 31 days)
- Create a separate aggregator measuring component that only aggregates interval measuring component type: 5 days types (horizon: 5 days, lag: 2 days)
- Create a third aggregator measuring component that aggregates the above aggregator measuring component types (horizon: 5 days, lag 31 days)

Expanding The Aggregation Horizon Periodically

Even with a 30-ish day aggregation lag, it's possible for an account with several months of estimated consumption to have its consumption "invalidated" when real readings arrive whose dial readings are less than the estimated readings (or when major retroactive data change takes place). Another similar situation is when a user manually corrects consumption after the aggregation horizon has moved forward.

To address situations like these, the use of ad-hoc aggregations with a user-specified aggregation horizon supports the notion of periodically (such as once a month or once a week) aggregating with a long aggregation horizon (up to 90 days) to catch as many retroactive changes as possible.



A Note About Using Non-Effective Dated Dimensions

Be aware that whenever aggregation takes place, the system uses master data that exists as of that date. For example, if an aggregation is performed using non effective-dated master data attributes (for example, aggregating by measuring component type) and a measuring component's type is changed, all days in the aggregation horizon will reflect the change regardless of when the user makes the change, because the measuring component type relationship is not effective-dated.

The impact of this depends on the nature of the change:

- If the master data was wrong, this provides a way to re-aggregate the data to accurately reflect the master data.
- If the change occurred effective as of a given date (but there's no effective dated attribute), this can result in incorrect aggregation values.

Because of this, it is recommended that where possible dimensions used in aggregations be effective-dated.

Aggregator Measuring Component Types

Aggregator measuring components have measuring component types just like all measuring components. In the case of aggregator measuring components, the measuring component type controls the aggregation period (horizon, lag, and cut off time) as well as the type of measuring components/items that are aggregated.

Aggregation Parameters

These attributes are used to define the aggregation period as described above, and include:

- Aggregation Horizon (in days): the number of days during which there's a potential change in measurement data for one or more of the measuring components associated an aggregator measuring component type.
- Aggregation Lag (in days): the number of days between the date on which aggregation calculations are performed and the end date of the aggregation period.
- Aggregation Cut Off Time (a time): the end time for aggregation calculations performed for aggregator measuring components of this type. This is used to ensure a consistent end time on all aggregation horizons and this is important if an aggregator is going to aggregate other aggregators.

Valid MC Types to Aggregate

Defines the valid measuring component types that can be included for aggregation calculations for aggregator measuring components of this type. Only measuring components whose types are defined here will be included in aggregation calculations for aggregator measuring components of this type. Note that any measuring component type (aggregator, physical, scratchpad, etc.) can be defined as long as the UOM of the identified value can be converted into the UOM of the aggregator measuring component's primary measurement value.

Valid Item Categories to Aggregate

Defines the valid item categories that will be included when aggregating measurements for aggregator measuring components of this type. Only items whose item types belong to the item categories defined will be included for aggregation calculations.

Value Identifiers

Defines value identifiers related to the current measuring component type. Value identifiers are used to provide short descriptions for the various types of values measured by the measuring component (KWH, KW, etc.).

Aggregator measuring components can derive up to 10 values in addition to the basic value (just like other measuring components). For example, the minimum, maximum, and average values could be stored on each measurement. The way the other values are derived is controlled by an enter algorithm on the Aggregate state.

Measuring Component Attributes

In addition the attributes on an aggregator measuring component's type, aggregator measuring components also have attributes that define how aggregations are performed. These include:

One Time Aggregation

Controls if the aggregator measuring component is used for one-time aggregations (used for ad hoc aggregations).

Consumption Aggregated Through Date Time

The date and time up to which consumption has been aggregated for the measuring component, based on the aggregation horizon used during the last aggregation calculation.

Next Aggregation Horizon

Defines the horizon for the next aggregation (either entered by a user or computed by the monitor algorithm when it determines it's time to aggregate).

Creating Aggregator Measuring Components

There are two ways in which aggregator measuring components are created:

- Automatically: The system will periodically create new aggregator measuring components when new "dimensional values" are detected. For example, if a new substation is added to the system, a new aggregator measuring component will be created for the substation.
- Manually: An end-user can create an aggregator measuring component at their discretion

Automatic Creation of Aggregator Measuring Components

The system automatically creates aggregator measuring components when it detects new instances of the dimension(s) being aggregated. This activity is referred to as dimension scanning. For example, if the system is configured to aggregate individual substations, a user will NOT have to manually set up an aggregator measuring component when a new substation is referenced on a service point. Rather, the system will create the new aggregator when it detects a substation on a service point that doesn't have an aggregator measuring component.

Similarly, if it is configured to aggregate distinct combinations of rate class (on usage subscription type) and postal code (on service point), the system will create an aggregator measuring component when it detects new combinations of rate class and postal code.

The periodic monitoring to ensure sufficient aggregation measuring components exist is implemented via monitoring algorithms on a related activity business object that references a corresponding measuring component type. This means that there must be a separate activity type / activity combination for every type of aggregation performed. These monitor algorithms include:

- A Monitor algorithm on the Active state that simply transitions the activity to the transitory Scan state.

Users can manually transition the activity to the Scan state by clicking the "Scan" button on the activity.
- An Enter algorithm on the Scan state finds the distinct combinations of dimensional values for this type of aggregation and checks that an aggregator measuring component that references the activity's aggregator measuring component type exists for every instance and creates new aggregator measuring components as appropriate. The algorithm also populates

the dimensional values on the new measuring components (mapped to searchable characteristics), and sets the Consumption Aggregated Through Date/Time field on the new aggregator measuring components to the current date and the Aggregation Cut Off Time on the aggregator measuring component type.

Hard-Wiring Service Types for Aggregations

If the dimensions used for aggregation do not include at least one service type-specific dimension (for example, in the case of aggregating by postal code only) AND the aggregation should not commingle measurements of different service types, it's important to:

- Declare the specific service type for the aggregation on the dimension scanner activity instance
- Declare the desired service type-specific measuring component types on the aggregator measuring component type

For example, if an electric and gas implementation aggregates consumption by postal code, they will need 2 activities: one will reference the Electric service type and the desired aggregator measuring component type, the other will reference the Gas service type and its aggregator measuring component type

The aggregator measuring components will have two dimensional attributes: postal code and service type.

Manual Creation Of Aggregator Measuring Components

Users can also create aggregator measuring component manually. This is useful when an ad hoc aggregation for an ad hoc time period is required, or when there is no activity that automatically creates instances of a given combination of dimensions.

To do this, the user simply clicks the “Add” link on the Aggregator Search zone title bar. The BPA script:

- Prompts the user to define the type of aggregation (this will present a list of measuring component types whose measuring component category is aggregator)
- Prompts the user to define the dimensional value(s), the desired aggregation time, and if this aggregation should be performed indefinitely or if it is a "one time"

Aggregation Algorithms

The base package includes aggregation algorithms that can populate columns on aggregated final measurements with the sum, max, min, average, and count of measurements in an interval. Additional algorithms can be developed by implementations teams to support other types of aggregations. For example, imagine a scenario where a user wants to see aggregated consumption during a critical peak period and contrast this to "normal" consumption. In this case, columns on the aggregated measurement could contain values like:

- Actual consumption
- “Normal” consumption
- The count of customers who reduced consumption between 0 and 10%
- The count of customers who reduced consumption between 10 and 25%
- The count of customers who reduced consumption more than 25%
- The count of customers who increased consumption between 0 and 10%
- The count of customers who increase consumption more than 10%

This scenario would necessitate the creation of a new aggregation algorithm that could piggy back on the base package version, and would only need additional logic to calculate each customer's "normal" consumption for each interval and populate the "counts" accordingly.

Base Package Aggregation Algorithms

The base contains the following algorithms used in aggregation processing.

Algorithm	Description
Aggregate Measurements of Aggregator's Constituent MCs (D2-AGG-MC)	Aggregates consumption for a set of measuring components (identified by the "Find Constituent Measuring Components" algorithm)
Aggregate Item Consumptions to Produce Max or Sum (D2-AGGITCMS)	Aggregates consumption for a set of items (identified by the "Find Constituent Items Based on Postal and Service Type" algorithm)
Find Constituent Measuring Components (D2-FIND-CMC)	Finds constituent measuring components.
Find Constituent Items Based on Postal and Service Type (D2-DETCITMPS)	Finds constituent items based on postal code and service type

Aggregating Specific Measuring Components

The above sections describe dimension-oriented aggregations, of the means of aggregating a variety of measuring components that are related to a given set of dimensional values (such as all measuring components for a given service type in a given postal code.

However, there is another class of aggregations where the requirements call for aggregating a set of specific measuring components. In addition, the aggregation formulae can be very unique, such as $(56.2\% \text{ of MC1} - 45.3\% \text{ of MC2}) * 1.034$.

Very specific aggregations of this should be implemented using usage rules, not aggregator measuring components.

Configuring Aggregations

This section provides high-level overviews of the steps involved in configuring custom aggregations. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

Configuring Custom Aggregations

Configuring custom aggregations involves the following steps:

1. Create a business object for the aggregator measuring component.

This will flatten the dimensional value(s) into searchable characteristics.

Whether this business object is a parent or a child of another aggregator business object depends on when periodic aggregation should occur:

- If you want the periodic aggregation to occur when another aggregation occurs, it can be a child business object (meaning that it inherits the lifecycle (and therefore the deferred monitor) of the parent)
- If you want to schedule its periodic aggregation independently from other aggregation business objects, this must NOT be a child business object as it will require its own deferred monitor (and deferred monitors can only be defined on parent business objects)

2. Create UI maps for the aggregator business object as follows:

- One to display the aggregator measuring component (Display)
- One to allow user to change / add a new one (Maintenance)

3. Create an info plug-in for the aggregator business object that concatenates together its dimension types and values.

4. Create a "Find Constituent Measuring Components" algorithm and plug it on the aggregator business object.

This will be passed the aggregator measuring component and the from and to date/times. It will insert the constituent measuring component IDs and the respective from / to date-time of each onto a temporary table.

5. Create a measuring component type instance and reference the new aggregator measuring component business object (as well as the types of constituent measuring component types that should be aggregated).
6. Create a query zone for Consumption Statistics search to allow users to find the aggregator measuring component.

Optionally:

7. Create a business object for the dimension scanner activity.

This should be a child business object of the base package dimension scanner business object.

8. Create UI maps for the activity business object, as follows:

- One to display the dimension scanner activity (Display)
- One to allow users to change / add a new one (Maintenance)

9. Create an info plug-in that will describe what it scans.

10. Create an Enter algorithm on the Scan state that finds distinct combinations of the dimensional values and creates new aggregator measuring components when new ones are detected.

You can reuse the base package deferred monitor batch control.

11. Create an activity type and reference the new dimension scanner business object.

Meter Data Management Base Package Aggregations

The Oracle Utilities Meter Data Management base package includes the following base package aggregators, delivered as aggregator measuring component business objects and measuring component types.

Aggregator	Description
Aggregator - Postal and Service Type (D2-Aggregator)	Aggregates measurements for measuring components based no postal code and service type dimensions.
Item Aggregator - Postal and Service Type (D2-ItemAggregatorPostalSvcType)	Aggregates measurements for items based no postal code and service type dimensions.

Business Intelligence Aggregators

The following aggregators are used by Oracle Utilities Meter Data Analytics, and aggregate measurement data (including quantities and counts) for constituent measuring components based on the following dimensions: Postal, City, Head-End, Device Type, Usage Calculation Group, Market and Service Provider, and Service Type.

Aggregator	Description
Measurement Measured Quantity Aggregator (D2-MeasuredQuantityAggregator)	Aggregates measurement quantities and distributes the constituents' measurements across Value Identifiers based on condition codes.
Measurement Quality Count Aggregator (D2-MsrmtQualityCountAggregator)	Aggregates measurement counts and distributes the constituents' counts across Value Identifiers based on condition codes.
Measurement Timeliness Count Aggregator (D2-MsrmtTimelinessCountAggr)	Aggregates measurement counts that arrived on time or are late.
Measurement Timeliness Quantity Aggregator (D2-MsrmtTimelinessQuantityAggr)	Aggregates measurement quantities that arrived on time or are late.

Note: When using the “dynamic determination of usage group” functionality, in which usage groups no longer need to be specified on the usage subscription directly and instead can be determined dynamically based on the customer's rate and device configuration type, the business intelligence aggregator logic uses a two-phase dimension scanning approach in which all dimensional value combinations with the exception of usage calculation group are determined, and a “sub-scanner” activity is created for each combination. Each sub-scanner is then processed to determine unique combinations consisting of its own dimensional values and the usage calculation groups valid for business intelligence, which should be configured using the “Dynamic Usage Group Mapping” extendable lookup (D2-UsageCalculationGroupLookup). The sub-scanners create new aggregator measuring components for any combination of attribute values for which an aggregator does not yet exist.

Sample Aggregator - Postal and Service Type

The Oracle Utilities Meter Data Management base package also includes an example aggregation that aggregates measurement quantities for constituent measuring components based on postal code and service type dimensions. The table below outlines the types of objects used in this aggregation, based on the steps outlined above), and the specific objects for each type.

Object Type	Base Package Example
Aggregator Measuring Component Business Object (Step 1)	D2-Aggregator (Aggregator - Postal and Service Type)
Aggregator Measuring Component UI Maps (Step 2)	Display: D2-AggMCDisp (Service Type and Postal Aggregator-Display) Maintenance: D2-AggMCMaint (Service Type and Postal Aggregator-Maintenance)
Aggregator Business Object Info Algorithm (Step 3)	D2-AMC-INFO (Service Type and Postal Aggregator - Information)
Find Constituent Measuring Components Algorithm (Step 4)	D2-DET-CMC (Find Constituent Measuring Components Based on Service Type and Postal)
Measuring Component Type (Step 5)	D2-AggregatorType (Aggregator Type)
Query Zone for Consumption Statistics Portal (Step 6)	D2-AGGMCQRY (Aggregator Search)
Dimension Scanner Activity Business Object (Step 7)	D2-ActivityAggDimScanner (Aggregator Creator - Postal / Service Type)
Dimension Scanner Activity UI Maps (Step 8)	Display: D2-AggDimScannerActDisp (Aggregator DS Activity-Display) Maintenance: D2-AggDimScannerActMaint (Aggregator DS Activity-Maintenance)
Dimension Scanner Activity Business Object Info Algorithm (Step 9)	D2-ADS-INFO (Aggregator Dimension Scanner Information)
Enter Algorithm for Scan State (Step 10)	D2-CRE-AGGMC (Aggregator MC Creation for Post Code and Service Type)
Activity Type (Step 11)	D2-ActivityTypeAggDimScanner (Aggregator Dimension Scanner Activity Type)

Chapter 14

Batch Processing

This chapter describes the base package batch processes provided with Oracle Utilities Service and Measurement Data Foundation and Oracle Utilities Meter Data Management. Use the Batch Control portal and Application Viewer for more details about these batch processes. This chapter includes:

- **Service and Measurement Data Foundation Batch Controls**
- **Meter Data Management Batch Controls**

Refer to the following documentation for more information about batch processing:

- Oracle Utilities Meter Data Management Batch Server Administration Guide
- Oracle Utilities Application Framework Business Process Guide (Batch Jobs)
- Oracle Utilities Application Framework Administration Guide (Defining Background Processes)

Service and Measurement Data Foundation Batch Controls

The table below lists the batch controls provided in the Service and Measurement Data Foundation base package.

Batch Control	Name / Description
D1-ACTVY	Activity Monitor Invokes the generic Application Framework auto-transition batch process that can do automatic/scheduled transition for activities
D1-CCMTR	Communication Comp State Monitor Process
D1-CEWR	Completion Event with Retry Monitor
D1-CMCS	Create Measurement Cycle Schedule Routes
D1-COMM	Batch Control for Communications
D1-CPSR	Complete Pending Msrmt Cycle Schd Routes - DEPRECATED
D1-CRERR	Command Request Error - Retry
D1-CRWT	Command Request Wait - Monitor
D1-CSPSR	Create SP Msrmt Cycle Schedule Rte Records
D1-DVEVS	Device Event Seeder Monitor Process
D1-DVEVT	Device Event MO Periodic Monitor Process
D1-EXTSC	Usage/Event Extract Scheduler Monitor
D1-GNIMD	Generic IMD Monitor - IMD Seeder
D1-IBCOM	Inbound Communication Wait - Monitor
D1-ICERR	Inbound Communication Error - Retry
D1-ICOMM	Generic Inbound Communication Batch Control
D1-IMD	IMD Monitor - Physical Devices Invokes monitoring rules associated with the current state of an initial measurement. All monitoring rules throughout the initial measurement's business object's inheritance chain are considered. Batch parameters govern whether the processing is further restricted by batch code, business object, status, etc. Enabling Sequence Handling Set the “Enable sequence handling” parameter (isSequenceHandlingEnabled) to “True” to specify that initial measurements for a measuring component be retrieved and ordered in ascending order by date time and create date time.
D1-IMDMC	IMD Monitor Unattached MCs

Batch Control	Name / Description
D1-INEVT	<p>Install Event MO Periodic Monitor Process</p> <p>Invokes monitoring rules associated with the current state of an install event. All monitoring rules throughout the install event's business object's inheritance chain are considered.</p> <p>By default, the process periodically monitors install events whose current state is not associated with a batch code. Batch parameters govern whether the processing is further restricted by batch code, business object and status.</p>
D1-MC	<p>MC MO Periodic Monitor Process</p> <p>Invokes monitoring rules associated with the current state of a measuring component. All monitoring rules throughout the measuring component's business object's inheritance chain are considered.</p> <p>By default, the process periodically monitors measuring components whose current state is not associated with a batch code. Batch parameters govern whether the processing is further restricted by batch code, business object and status.</p>
D1-MSRMT	<p>Measurement - Derive Other Values</p> <p>Initiates a reprocessing of Derive Other Values for Measurements. To do this, it moves the status of the Measurement from OK to REDERIVEVAL. The REDERIVEVAL status will have a monitor algorithm that will execute all "Derive Other Values" algorithms that have been attached to the business object's Plug-in spot.</p> <p>Batch parameters are required and are used to select the specific business object as well as the starting date and ending date for the records to be processed.</p>
D1-OBCOM	Outbound Communication Wait - Monitor
D1-OCERR	Outbound Communication Error - Retry
D1-OCOMM	Generic Outbound Communication Batch Control
D1-OCWT	Outbound Communication Wait - Monitor
D1-PMCS	DEPRECATED - Process Measurement Cycle Schedule
D1-PSACC	Payload Statistics Accumulation - Monitor
D1-PSPSR	Process SP / MC Schedule Route Records
D1-RMCRE	<p>Related MC Consumption Sync</p> <p>This batch control invokes the generic Application Framework auto-transition batch process to transition consumption synchronization activities. This job is intended to process consumption synchronization activities that are in the Pending status.</p>
D1-RMCRR	<p>Related MC Consumption Sync - Retry Activities</p> <p>This batch control invokes the generic Application Framework auto-transition batch process to transition consumption synchronization activities. This job is intended to process consumption synchronization activities that are in the Error status.</p>
D1-SMMTR	Smart Meter State Monitor Process

Batch Control	Name / Description
D1-SP	<p>Service Point MO Periodic Monitor Process</p> <p>Invokes monitoring rules associated with the current state of a service point. All monitoring rules throughout the service point's business object's inheritance chain are considered.</p> <p>By default, the process periodically monitors service points whose current state is not associated with a batch code. Batch parameters govern whether the processing is further restricted by batch code, business object and status.</p>
D1-SYNC	Data Sync MO Periodic Monitor Process
D1-TOU	<p>Time of Use Map Data Generation Monitor</p> <p>Invokes monitor algorithms for Time Of Use Map instances whose current state does not reference a monitor process.</p>
D1-UT	Usage Transaction Monitor Process - NOT USED
D1-UTCD	Usage Transaction Calculate Defer Monitor - NOT USED
D1-UTCRN	Usage Transaction Correction Monitor
D1-UTID	Usage Transaction Issue Detected Monitor - NOT USED
D1-UTSED	Usage Transaction Seeder - Error - NOT USED

Synchronization Request Batch Controls

The table below lists the batch controls used by the Service and Measurement Data Foundation for data synchronization. “Initial Sync Request - Load Data <batch control>” batch controls load data (created new instances of business objects) for requests of the appropriate type (device, measuring component, etc.). “Initial Sync Request - Resolve Keys <batch control>” batch controls invoke a generic maintenance object transition process to invoke the “Resolve Keys - Initial Sync” algorithm for synchronization requests of the appropriate type.

Batch Control	Description
D1-CMSYN	Composite Sync Request Transition Composite Sync Request BO instances in Pending state to the next state.
D1-SIHER	Initial Sync Request - Error Transition Initial Inbound Sync Request BO instances in Error state to the next state.
D1-SILCN	Initial Sync Request - Load Data Contact
D1-SILDC	Initial Sync Request - Load Data DC
D1-SILDV	Initial Sync Request - Load Data Device
D1-SILIE	Initial Sync Request - Load Data IE
D1-SILMC	Initial Sync Request - Load Data MC
D1-SILSP	Initial Sync Request - Load Data SP
D1-SILUS	Initial Sync Request - Load Data US
D1-SIKCN	Initial Sync Request - Resolve Keys Contact
D1-SIKDC	Initial Sync Request - Resolve Keys DC
D1-SIKDC	Initial Sync Request - Resolve Keys Device
D1-SIKIE	Initial Sync Request - Resolve Keys IE
D1-SIKMC	Initial Sync Request - Resolve Keys MC
D1-SIKSP	Initial Sync Request - Resolve Keys SP
D1-SIKUS	Initial Sync Request - Resolve Keys US
D1-SIOER	Ongoing Sync Request -Error Transition Ongoing Inbound Sync Request BO instances in Error state to the next state.
D1-SIOPE	Ongoing Sync Request - Pending Transition Ongoing Inbound Sync Request BO instances in Pending state to the next state.
D1-SRSDE	Sync Request Seeder - Error Transition Sync Request Seeder BO instances in Error state to the next state.
D2-SIKUS	Initial Sync Request - Resolve Keys US
D2-SILUS	Initial Sync Request - Load Data US

Service and Measurement Data Foundation Batch Processing Guidelines

This section provides some guidelines around how to schedule batch processing for the batch controls used with Service and Measurement Data Foundation (including those used by Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway). Note that there are no strict dependencies between batch controls, meaning there are no situations in which you must run a specific batch process before running any other. Dependencies are based on the way a utility wants to run their business.

Ongoing Processes

Throughout the course of the day, utilities will likely want to run jobs to bring measurement and event data in from their various metering / head-end systems and other external systems. To accompany this, they might want to run the following batch processes:

Ongoing Master Data Sync Processing

If the system is integrated with a customer information system such as Oracle Utilities Customer Care and Billing and synchronizing master data such as service points, install events, usage subscriptions, etc. on an ongoing basis, this data should be as up-to-date as possible when loading and processing data. This can include:

- Processing composite sync requests that will create individual sync requests for multiple objects (D1-CMSYN)
- "Cleaning out" any ongoing sync requests in error from previous runs (D1-SIOER) and those that have more severe issues and remain as sync request "seeders" (D1-SRSDE)
- Processing sync records that are pending (D1-SIOPE), and again those in error (D1-SIOER).

Note: Processing records in error after processing the batch of "pending" ongoing sync requests is a recommended practice as it serves to clean up dependent sync requests that may have been picked up in an incorrect order. For example, attempting to process a Usage Subscription sync request referencing a Service Point before processing that service point's sync request will put the usage subscription sync request in error. Running D1-SIOER will pick up that usage subscription sync request that had been left behind, and can process it successfully now that the service point it references has been synced in.

Ongoing master data sync could be part of a nightly schedule rather than running it throughout the day, depending on whether the attributes being synced impact VEE and/or usage processing. If nothing synced from a source system impacts VEE, then sync processing need not be run before VEE.

Command Processing

If using Oracle Utilities Smart Grid Gateway to process device commands, it's important to keep communications flowing to and from smart meters to provide the most accurate picture of the state of a given meter. This would include:

- Retrying inbound communications in error (D1-ICERR)
- Retrying outbound communications in error (D1-OCERR)
- Processing outbound communications waiting for a response (D1-OCWT) to see if they should be timed out.
- Processing command request activities in error (D1-CRERR) and those that are waiting (D1-CRWT)

Note that base package business objects for communications and command activities are designed to trap any processing errors encountered and transition the object into an Error state. To deal with unexpected errors that can't be trapped, which could leave communications / command activities in unmonitored states, implementations can choose to configure their own batch controls based on the delivered D1-ACTVY and D1-COMM batch controls, restricting records processed by business object or maintenance object as needed.

Initial Measurement Processing

For initial measurement processing, the following batch processes should be scheduled on an ongoing basis:

- Processing initial measurements in the pending state (D1-IMD)
- Processing initial measurements in the exception state (D1-GNIMD)

In addition, processing initial measurement seeders (IMD Seeders) in error should be performed. This can be accomplished through creation of a custom batch control that references the following Java program used by the “Generic IMD Monitor” (D1-GNIMD):

```
com.splwg.base.domain.common.businessObject.batch.AutoTransitionBatchProcess
```

This custom batch process could be configured to restrict processing to the IMD Seeder business object (D1-IMDSeeder).

Event Processing

The base package is configured such that device events are processed immediately upon receipt, since they might need to be sent to some other application such as an outage system. This can be changed by configuring a monitor process on the device event business object to stop records in a specified state, and then use a batch process to process the events all at once. Beyond this type batch-oriented processing for events, other even processing could include:

- Re-processing device event "seeders" in error (D1-DVEVS),
- Picking up device events for processing if they've stopped in any state (D1-DVEVT).

If device events are configured to be held from being sent onto downstream applications, such as to prevent "flicker" outage events (an outage event and a restore event received in rapid succession) from being sent, device event monitoring (D1-DVEVT) should be set up to be run periodically to ensure timely transmission of events.

Daily/Nightly

In addition to the above ongoing processes, the following daily or nightly processes can also be scheduled.

Consumption Synchronization

Consumption synchronization is driven by monitoring consumption synchronization activities via the “Related MC Consumption Sync” and “Related MC Consumption Sync - Retry Activities” batch controls (D1-RMCRE and D1-RMCRR). These batch controls transition the activities to the "Re-Estimate" state, at which point "Enter" algorithms ("Interval MC Consumption Sync" and/or "Scalar MC Consumption Sync") perform consumption synchronization processing.

Note: This process should be executed prior to Periodic Estimation to eliminate any chance of additional estimated measurements being generated that coincide with a period that is being synchronized. This will ensure no measurements are generated by Periodic Estimation and then immediately estimated once more by this job.

Periodic Interval/Smart Meter Estimation

Periodic estimation is driven by monitoring devices via the “Smart Meter State Monitor Process” batch control (D1-SMMTR) which executes a monitor algorithm to execute the estimation algorithm on the measuring component business object, which in turn can be configured to create Estimated initial measurements.

Measurement Cycle Processing (Optional)

If using a billing system that doesn't request for bill determinants and requires that bill determinants be pushed to it, the following processes can be used:

- Creating meter read cycle schedule routes (D1-CMCS)
- Creating SP / Measurement Cycle Schedule Routes (D1-CSPSR)

- Processing SP / Measurement Cycle Schedule Routes (D1-PSPSR)

The above processes can also be used to drive the download of meter read cycle & route information for manually-read meters.

Periodic / Ad Hoc

In addition to the above ongoing and daily/nightly processes, the following periodic processes can be run on an as needed basis.

TOU Map Data Generation

TOU map data must be in place for all TOU maps used in usage calculations. Generation of this data is performed using the “Time of Use Map Data Generation Monitor” batch control (D1-TOUTR). This process can be performed for long time periods, such as a year, generating data for all time-of-use maps for the entire following year, or it could be done more frequently, such as whenever schedules are updated via the TOU map templates.

Re-Derive Measurement Values

If certain data such as a factor value was found to be incorrect, derived measurement values might need to be re-calculated across all final measurements for a given date range. This can be performed using the “Measurement - Derive Other Values” batch control (D1-MSRMT). Given the possible volume of data impacted by this process, careful consideration should be given before performing this process.

Master Data Monitoring

The base package also provides batch processes intended to monitor service points (D1-SP), install events (D1-INEVT), and measuring components (D1-MC) on an ad-hoc basis if processing driven from the lifecycles of these objects is needed. Note that the base package contains no such processing.

Meter Data Management Batch Controls

The table below lists the batch controls provided in the Oracle Utilities Meter Data Management base package.

Batch Control	Name / Description
D2-ADS	Aggregator Dimension Scanner Monitor
D2-AGG	Aggregation Monitor
D2-SIKUS	Initial Sync Request - Resolve Keys US
D2-SILUS	Initial Sync Request - Load Data US
D2-UT	Usage Transaction Monitor Process
D2-UTCD	Usage Transaction Calculate Defer Monitor
D2-UTEA	Usage Transaction by External Account ID
D2-UTID	Usage Transaction Issue Detected Monitor
D2-UTSED	Usage Transaction Seeder - Error
D2-VEEDF	VEE Exception Monitor

Meter Data Management Batch Processing Guidelines

In addition to the batch processes outlined for the Service and Measurement Data Foundation above, Oracle Meter Data Management can employ the following ad-hoc or periodic batch processes.

Usage Transaction processing

If Oracle Utilities Meter Data Management is integrated with a billing system (such as Oracle Utilities Customer Care and Billing) that requests bill determinants, usage transaction processing should be coordinated with that billing system's billing process. This can include:

- Reprocessing usage transaction "seeders" in error (D2-UTSED)
- Reprocessing usage transactions in error (D2-UTID)

Requests from a billing system that have been indicated as "batch requests" (such as those produced by Oracle Utilities Customer Care and Billing batch billing process) accumulate in a "calculation deferred" state to be processed specially by the "Usage Transaction Calculate Defer Monitor" batch control (D2-UTCD).

If unexpected errors occur that leave usage transactions in an unmonitored state, the "Usage Transaction Monitor" batch control (D2-UT), or one based on this batch control with parameter values tailored to any specific requirements, can be used to process those usage transactions.

Aggregation

Aggregation calculations should be run on an as needed basis. This can include:

- Scanning for new aggregation dimension (D2-ADS)

This process is applicable if the system is configured to use aggregation dimension scanners to detect new aggregation dimensions (such as a service point referencing a new transformer for which an aggregator measuring component doesn't currently exist)

- Performing aggregation calculations (D2-AGG)

Note that aggregation calculations should precede usage transaction processing if aggregated values serves as input to the calculation of bill determinants.

Meter Data Management Business Intelligence Batch Controls

The table below lists the batch controls used with Oracle Utilities Meter Data Management Business Intelligence.

Batch Control	Name / Description
D1-ACSDX	Activity BO Status/Reason Dim Extract
D1-ACSIL	Activity BI Status/Reason Initial Load
D1-ACTFX	Activity Accumulation Fact Extract
D1-ACTL	Activity Accumulation Fact Initial Load
D1-ADRDY	Address Dimension Extract
D1-ATYDX	Activity Type Dimension Extract
D1-ATYIL	Activity Type Dimension Initial Load
D1-DESDX	Device Evt BO Status/Reason Dim Extract
D1-DESIL	Device Evt BO Status/Reason Initial Load
D1-DETDY	Device Event Type Dimension Extract
D1-DETIL	Device Event Type Initial Load
D1-DEVFX	Device Event Accumulation Fact Extract
D1-DEVL	Device Event Initial Load
D1-DVCDX	Device Dimension Extract
D1-DVCIL	Device Dimension Initial Load
D1-IESDX	Install Evt BO Status/Reason Dim Extract
D1-IESIL	Install Evt BO Status/Reason Initial Load
D1-INEFX	Install Event Fact Extract
D1-INEIL	Install Event Initial Load
D1-LNMDX	Days Since Last Normal Measurement Extract
D1-LNML	Days Since Last Normal Msrmt Initial Load
D1-MCDX	Measuring Component Dimension Extract
D1-MCIL	Measuring Component Dimension Initial Load
D1-SPAFX	Service Point Accumulation Fact Extract
D1-SPDX	Service Point Dimension Extract
D1-SPIL	Service Point Initial Load
D1-SPRDY	Service Provider Dimension Extract
D1-SPRIL	Service Provider Initial Load

Batch Control	Name / Description
D1-SPSDL	Service Point Snapshot Download
D1-SPSDX	SP BO Status/Reason Dimension Extract
D1-SPSFX	Service Point Snapshot Fact Extract
D1-SPSIL	SP BO Status/Reason Initial Load
D2-CONDX	Contact Dimension Extract
D2-CONIL	Contact Initial Load
D2-CSTDx	Usage Snapshot Type Dimension Extract
D2-CSTIL	Usage Snapshot Type Initial Load
D2-DUGDL	Usage Subscription Usage Group Download
D2-EXLIL	Exception Severity Lookup Extract
D2-EXTDX	Exception Type Dimension Extract
D2-EXTIL	Exception Type Initial Load
D2-ITLIL	IMD Type Lookup Extract
D2-LUTDX	Days Since Last UT Dimension Extract
D2-LUTIL	Days Since Last UT Initial Load
D2-MRCDX	Measurement Condition Dimension Extract
D2-MRCIL	Measurement Condition Initial Load
D2-MVREF	Materialized View Refresh
D2-SPCFX	Usage Snapshot Fact Extract
D2-SUAFX	Unreported Usage Analysis Fact Extract
D2-SVEFX	VEE Exception Snapshot Fact Extract
D2-USGDL	Service Point Usage Snapshot Download
D2-UGDX	Usage Group Dimension Extract
D2-UGIL	Usage Group Initial Load
D2-USDX	Usage Subscription Dimension Extract
D2-USIL	Usage Subscription Initial Load
D2-UTADX	Unreported Usage Analysis Type Dim Extract
D2-UTAIL	SP UT Aging Snapshot Type Initial Load
D2-UTIL	UOM/TOU Dimension Extract
D2-UTSIL	UOM/TOU/SQI Dimension Extract
D2-UUSDL	SP Unreported Usage Snapshot Download
D2-VEEDL	SP VEE Exception Snapshot Download
D2-VERDX	VEE Rule Dimension Extract

Batch Control	Name / Description
D2-VERIL	VEE Rule Dimension Initial Load

Refer to the *Oracle Utilities Meter Data Management Business Intelligence Data Mapping Guide* for more information for more information about setting up batch processing for Oracle Utilities Meter Data Management Business Intelligence.

Chapter 15

Integrating Oracle Utilities Meter Data Management with Other Systems

This chapter provides information related to integrating Oracle Utilities Meter Data Management with other applications, including:

- **Integrating with a Customer Information System**
- **Integrating with Oracle Utilities Operational Device Management**
- **Integrating with Oracle Utilities Analytics**
- **Integrating with Oracle Utilities Customer Self Service**
- **Oracle Utilities DataConnect**
- **Initial Measurement and Device Event XML Formats**
- **Usage and Event Import - Message Driven Bean Configuration**

Integrating with a Customer Information System

This section provides an overview of how Oracle Utilities Meter Data Management supports integrations with a customer information system.

In an integration between Oracle Utilities Meter Data Management and a customer information system such as 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 (the customer information system) 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 two core business processes:

- **Data Synchronization**
- **Processing Usage Transaction Requests**

Data Synchronization

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, customer, or service point-related data. Synchronizing this data between the two systems ensures that all account, customer, and service point-related data in Oracle Utilities Meter Data Management is correct and up to date before usage transaction calculations are performed. This synchronization process is supported through a set of business objects, master configurations, batch controls, and pre-configured XAI Inbound Services.

Types of Requests

Data synchronization is performed via synchronization requests sent from Oracle Utilities Customer Care and Billing via a middleware integration component. Oracle Utilities Meter Data Management supports three types of synchronization requests:

Initial Synchronization Requests

Initial synchronization requests are used when initially setting up Oracle Utilities Meter Data Management. They facilitate import of data that creates devices, device configurations, measuring components, service points, install events, contacts, and usage subscriptions in Oracle Utilities Meter Data Management based on corresponding data in Oracle Utilities Customer Care and Billing.

Ongoing Synchronization Requests

Ongoing synchronization requests are used when updating existing data in Oracle Utilities Meter Data Management based on changes in corresponding data in Oracle Utilities Customer Care and Billing. Ongoing synchronization requests can be used to update contacts, devices, device configurations, measuring components, install events, service points, and usage subscriptions.

Composite Synchronization Requests

Composite synchronization requests are requests that contain synchronization requests for multiple types of data within a single request. For example, a composite request could contain requests to update device, device configuration, measuring component, and install event data. This supports situations where multiple types of data must be updated based on a single change in Oracle Utilities Customer Care and Billing.

Base Package Synchronization Request Business Objects

The table below lists the base package synchronization request business objects used by Oracle Utilities Meter Data Management.

Business Object	Description
D1-CompositeSyncRequest	<p>Composite Sync Request</p> <p>Used as a parent business object to define the behavior of a composite synchronization request that contains multiple synchronization requests (defined as child business objects). It is designed to accept a sync request whose information needs to be propagated to multiple maintenance objects. Its children business objects contain the elements for all the maintenance objects involved and then, within their processing, create a sync request for each maintenance object instance required by the inbound request.</p> <p>For example, composite synchronization requests that contains requests for device, device configuration, and install event could use a composite business object with the Device, Device Configuration, and Install Event synchronization request business objects as child business objects. When processed, this business object would create instances of the Device, Device Configuration, and Install Event synchronization request business objects.</p>
D1-CompositeSyncRequestDC	<p>Device Config Composite Sync Request</p> <p>An example composite synchronization request. This business object a child of the D1-CompositeSyncRequest business object.</p>
D1-InitialSyncRequestContact	<p>Contact Initial Sync Request</p> <p>Instances of this business object represent individual initial contact synchronization requests.</p>
D1-InitialSyncRequestDC	<p>Device Configuration Initial Sync Request</p> <p>Instances of this business object represent individual initial device configuration synchronization requests.</p>
D1-InitialSyncRequestDevice	<p>Device Initial Sync Request</p> <p>Instances of this business object represent individual initial device synchronization requests.</p>
D1-InitialSyncRequestIE	<p>Install Event Initial Sync Request</p> <p>Instances of this business object represent individual initial install event synchronization requests.</p>

Business Object	Description
D1-InitialSyncRequestMC	Measuring Component Initial Sync Request Instances of this business object represent individual initial measuring components synchronization requests.
D1-InitialSyncRequestSP	Service Point Initial Sync Request Instances of this business object represent individual initial service point synchronization requests.
D1-OngoingSyncReqAckMsg	Ongoing Sync Request Acknowledgement Used to send a message to the sending system to acknowledge receipt of an ongoing synchronization request.
D1-OngoingSyncReqScalarMtrRead	Scalar Meter Read Ongoing Sync Request Instances of this business object represent individual ongoing scalar meter read synchronization requests.
D1-OngoingSyncRequestContact	Contact Ongoing Sync Request Instances of this business object represent individual ongoing contact synchronization requests.
D1-OngoingSyncRequestDC	Device Configuration Ongoing Sync Request Instances of this business object represent individual ongoing device configuration synchronization requests.
D1-OngoingSyncRequestDevice	Device Ongoing Sync Request Instances of this business object represent individual ongoing device synchronization requests.
D1-OngoingSyncRequestIE	Install Event Ongoing Sync Request Instances of this business object represent individual ongoing install event synchronization requests.
D1-OngoingSyncRequestMC	Measuring Component Ongoing Sync Request Instances of this business object represent individual ongoing measuring component synchronization requests.
D1-OngoingSyncRequestSP	Service Point Ongoing Sync Request Instances of this business object represent individual ongoing service point synchronization requests.
D1-SeederSyncMasterConfig	Seeder Sync Request Master Configuration Used to define the foreign keys for each entity being synchronized and their corresponding views that will be used in translating the external (old) key to production (new) key references during initial conversion processing or ongoing synchronization processing.

Business Object	Description
D1-SyncRequestSeeder	Sync Request Seeder Used to identify the appropriate synchronization business object to create when processing synchronization requests.
D1-SynchronizationAddContact	Contact Synchronization Add Used when adding a new contact as a result of a synchronization request.
D1-SynchronizationAddDC	Device Configuration Synchronization Add Used when adding a new device configuration as a result of a synchronization request.
D1-SynchronizationAddDevice	Device Synchronization Add Used when adding a new device as a result of a synchronization request.
D1-SynchronizationAddIE	Install Event Synchronization Add Used when adding a new install event as a result of a synchronization request.
D1-SynchronizationAddMC	Measuring Component Synchronization Add Used when adding a new measuring component as a result of a synchronization request.
D1-SynchronizationAddSP	Service Point Synchronization Add Used when adding a new service point as a result of a synchronization request.
D2-InitialSyncRequestUS	Usage Subscription Initial Sync Request Instances of this business object represent individual initial usage subscription synchronization requests.
D2-OngoingSyncRequestUS	Usage Subscription Ongoing Sync Request Instances of this business object represent individual ongoing usage subscription synchronization requests.
D2-SynchronizationAddUS	Usage Subscription Synchronization Add Used when adding a new usage subscription as a result of a synchronization request.

Master Configurations

Master configurations are used to define aspects of the synchronization process, including resolution of foreign keys and the type of synchronization business objects to use for each type of data being synchronized.

The table below lists the master configurations used in data synchronization processing.

Master Configuration	Description
Master Data Synchronization Configuration	Lists all foreign key references that need resolution. Each one should reference the view that contains the external key / production key cross-reference. For entities that undergo both the initial and the ongoing sync, two views are specified. For entities that undergo the ongoing sync, an external system / ID type mapping is specified to cater for entities that might be synchronizing from more than one external system.
Seeder Sync Request Master Configuration	Lists the maintenance objects (device, device configuration, etc.) that require synchronization. Each references the synchronization business object that needs to be instantiated when processing a synchronization request for that maintenance object. For maintenance objects that undergo both initial and the ongoing synchronization, two business objects are specified.

Refer to the Oracle Utilities Application Framework documentation for more information about master configurations.

Batch Controls

Batch controls perform processing for initial synchronization requests such as allocating keys to data, resolving foreign keys, and loading data (instantiating business objects representing entities such as devices, measuring components, etc.).

“Initial Sync Request - Resolve Keys XXX” batch controls invoke a generic maintenance object transition process to invoke the “Resolve Keys - Initial Sync” algorithm for synchronization requests of the appropriate type. Parameters used by “resolve keys” batch controls include:

- **Maintenance Object:** (Required) the maintenance object (device, device configuration, etc.) to be processed. This must be set to the Sync Request maintenance object for the batch control (device for device synchronization requests, service point for service point synchronization requests, etc.)
- **Restrict By Batch Code:** Restricts processing to synchronization requests whose current state is linked to this batch code.
- **Restrict By Business Object:** Restricts processing to synchronization requests linked to this business object.
- **Restrict By Status Code:** Restricts processing to synchronization requests of this status (default: KEY_ALLOCATD).
- **Max Errors:** Specifies the maximum number of errors allowed before the process exits.

“Initial Sync Request - Load Data XXX” batch controls load data (created new instances of business objects) for requests of the appropriate type (device, measuring component, etc.). Parameters used by “load data” batch controls include:

- **Maintenance Object:** (Required) the maintenance object (device, device configuration, etc.) to be processed. This must be set to the Sync Request maintenance object for the batch control (device for device synchronization requests, service point for service point synchronization requests, etc.)
- **Restrict By Batch Code:** Restricts processing to synchronization requests whose current state is linked to this batch code.
- **Restrict By Business Object:** Restricts processing to synchronization requests linked to this business object.
- **Max Errors:** Specifies the maximum number of errors allowed before the process exits.

The table below lists the batch controls used by initial synchronization requests

Batch Control	Description
D1-CMSYN	Composite Sync Request Transition Composite Sync Request BO instances in Pending state to the next state.
D1-SIIER	Initial Sync Request - Error Transition Initial Inbound Sync Request BO instances in Error state to the next state.
D1-SILCN	Initial Sync Request - Load Data Contact
D1-SILDC	Initial Sync Request - Load Data DC
D1-SILDV	Initial Sync Request - Load Data Device
D1-SILIE	Initial Sync Request - Load Data IE
D1-SILMC	Initial Sync Request - Load Data MC
D1-SILSP	Initial Sync Request - Load Data SP
D1-SILUS	Initial Sync Request - Load Data US
D1-SIKCN	Initial Sync Request - Resolve Keys Contact
D1-SIKDC	Initial Sync Request - Resolve Keys DC
D1-SIKDC	Initial Sync Request - Resolve Keys Device
D1-SIKIE	Initial Sync Request - Resolve Keys IE
D1-SIKMC	Initial Sync Request - Resolve Keys MC
D1-SIKSP	Initial Sync Request - Resolve Keys SP
D1-SIKUS	Initial Sync Request - Resolve Keys US
D1-SIOER	Ongoing Sync Request -Error Transition Ongoing Inbound Sync Request BO instances in Error state to the next state.
D1-SIOPE	Ongoing Sync Request - Pending Transition Ongoing Inbound Sync Request BO instances in Pending state to the next state.

Batch Control	Description
D1-SRSDE	Sync Request Seeder - Error Transition Sync Request Seeder BO instances in Error state to the next state.
D2-SIKUS	Initial Sync Request - Resolve Keys US
D2-SILUS	Initial Sync Request - Load Data US
F1-SAKRQ	Sync Request Allocate Keys Monitor Allocates keys to the entities that needs to be added. Batch parameters govern whether the processing is further restricted by batch code, business object, maintenance object and max errors.
F1-SRLRQ	Sync Request Load Records Monitor Performs fast inserts using batch add.
F1-SYNRQ	Sync Request Monitor Process Invokes monitoring rules associated with the current state of sync requests. All monitoring rules throughout the sync request's business object's inheritance chain are considered. Batch parameters govern whether the processing is further restricted by batch code, business object and status.
F1-SYSRQ	Sync Request Sampling Monitor (Deferred) Invokes monitoring rules associated with the current state of inbound synchronization requests. All monitoring rules throughout the inbound synchronization request's business object's inheritance chain are considered.

Batch Control Scheduling

The following table specifies the order in which the batch controls on the Initial Sync Request BO life cycle should be executed. The first row identifies the maintenance object for which the synchronization request is intended and the first column specifies the type of process.

	Contact	US	SP	Install Event	Device Config	Device	Measuring Component
Transformation / Schema Validation	1	1	1	1	1	1	1
Allocate Keys	2	2	2	2	2	2	2
Foreign Key Resolution /BO Validation	3	8	3	6	3	3	3
Load Data	4	9	5	7	4	4	4

Note that before the Key Resolution job is run, all the Key Allocation Jobs need to finish. This ensures that all foreign key references can be subsequently resolved.

Some business object-level validation is dependent on other entities being completely loaded first. The sequence numbers above allow for this. For example, usage subscriptions business object validation is dependent on service points existing; Install Event business object validation is dependent on both service points and devices existing.

XAI Inbound Services

XAI inbound services are used to facilitate invoking the Sync Request Seeder business object by the middleware components upon receipt of a synchronization request.

The table below lists the pre-configured XAI Inbound Services used to process synchronization requests sent from Oracle Utilities Customer Care and Billing.

XAI Inbound Service	Description	Schema Name
D1-SyncRequestInbound	Sync Request Inbound	D1-SyncRequestSeeder (BO)
D1-SyncRequestInboundComposite	Sync Request Inbound Composite	D1-CompositeSyncRequestDC (BO)

Refer to the Oracle Utilities Application Framework documentation for more information about XAI inbound services.

Understanding Synchronization Request Processing

This section provides an overview of the processing that takes place when a synchronization request is sent. For each step in the process, the table below provides a brief description of the processing that takes place, and lists the specific objects involved.

Step	Process	Objects
1.	Oracle Utilities Customer Care and Billing sends a synchronization request to the middleware integration layer. For example, consider a request to update information about a service point.	
2.	The middleware components transforms the request from the Customer Care and Billing format, to the format used by Oracle Utilities Meter Data Management (this format is based on the business object schemas of the synchronization request business objects).	
3.	The middleware component invoke the appropriate XAI Inbound Service, and sends the transformed request.	XAI Inbound Service: D1-SyncRequestInbound (mapped to the D1-SyncRequestSeeder business object)
4.	The XAI Inbound Service invokes the Sync Request Seeder business object, which in turn, determines which synchronization request business object to create (based on the type of data in the synchronization request and the Seeder Sync Master Configuration).	Synchronization Request BO: D1-OngoingSyncRequestSP
5.	For initial synchronization requests, background processing creates master data for each synchronization request, including the following steps: <ul style="list-style-type: none"> Data Transformation / Schema Validation Allocate Keys Resolve Foreign Keys / Validate Business Object Load Data See Batch Control Scheduling on page 15-8 for more information about scheduling batch processing for initial synchronization requests.	

Step	Process	Objects
6.	For ongoing synchronization requests, the following steps are performed by Enter algorithms on the synchronization request business object lifecycle states: <ul style="list-style-type: none"> Data Transformed/Basic Schema Validated: <ul style="list-style-type: none"> Determine Sync Request BO Setup Transformed Data Pre-Added: <ul style="list-style-type: none"> Sync Request Pre-Add Data FKs Resolved: <ul style="list-style-type: none"> Resolve Keys - Ongoing Sync Updating: <ul style="list-style-type: none"> Sync Request Update Data 	Setup Transformed Data Algorithm: D1-SETTRANDT Sync Request Pre-Add Data Algorithm: D1-SR-PREADD Resolve Keys - Ongoing Sync Algorithm: D1-RESKEYFAL Sync Request Update Data Algorithm: D1-SR-UPDDAT

Processing Usage Transaction Requests

Oracle Utilities Customer Care and Billing uses bill determinant data (usage transactions 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 usage transaction calculations to Oracle Utilities Meter Data Management, which in turn performs the requested calculations, and publishes the results back to Oracle Utilities Customer Care and Billing. Processing usage transaction requests is supported through a set business objects, pre-configured XAI Inbound services, and processing methods.

Business Objects

The table below lists the business objects used when processing usage transaction requests.

Business Object	Description
D2-UsgTranSeeder	Usage Transaction Seeder Used to determine the usage transaction business object to use when creating new usage transactions.

XAI Inbound Services

XAI inbound services are used to facilitate invoking the Usage Transaction Seeder business object by the middleware components upon receipt of a usage transaction request.

The table below lists the pre-configured XAI Inbound Services used to process usage transaction requests sent from Oracle Utilities Customer Care and Billing.

XAI Inbound Service	Description	Schema Name
D2-UsageTransactionRequestInbound	Usage Transaction Request Inbound	D2-UsgTranSeeder (BO)

Processing Methods

Processing methods are used to determine the usage transaction business object to use when creating usage transactions based on the requests, and for determining the method by which usage transactions are sent back to Oracle Utilities Customer Care and Billing.

Processing Method	Description
D2-HowToSendUSInfoOnline	How To Send US Information - Online Used to specify the method (batch control, business object, or outbound message) by which usage transactions are sent to subscribing systems in real time.
D2-HowToSendUSInformation	How To Send US Related Information Used to specify the method (batch control, business object, or outbound message) by which usage transactions are sent to subscribing systems.
D2-HowToCreateUSInformation	How To Create US Related Information Used to determine the type of usage transaction business object to create.
D2-HowToSendUSInfoBatch	How To Send US Related Information Batch Used to specify the method (batch control, business object, or outbound message) by which usage transactions are sent to subscribing systems via batch processing.

Understanding Usage Transaction Request Processing

This section provides an overview of the processing that takes place when a usage transaction request is sent. For each step in the process, the table below provides a brief description of the processing that takes place, and lists the specific objects involved.

Step	Process	Objects
1.	Oracle Utilities Customer Care and Billing sends a usage transaction request to the middleware integration layer.	
2.	The middleware components transforms the request from the Customer Care and Billing format, to the format used by Oracle Utilities Meter Data Management (this format is based on the business object schemas of the synchronization request business objects).	
3.	The middleware component invoke the appropriate XAI Inbound Service, and sends the transformed request.	XAI Inbound Service: D2-UsageTransactionRequestInbound (mapped to the D2-UsgTranSeeder business object)

Step	Process	Objects
4.	<p>The XAI Inbound Service invokes the Usage Transaction Seeder business object. This business object:</p> <ul style="list-style-type: none"> Determines the usage subscription ID based on an external usage subscription ID. This processing is performed via a Pre-Processing algorithm. Determines the appropriate usage transaction business object to create. This processing is performed via a Pre-Processing algorithm and a processing method defined for the “Usage Transaction Creation” processing role on the service provider that represents the external system. 	<p>Pre-Processing Algorithm: D2-DETUSID (Determine Usage Subscription ID)</p> <p>Pre-Processing Algorithm: D2-DETUTBO (Determine Usage Transactions Business Object)</p> <p>Processing Method: D2-HowToCreateUSInformation (How To Create Usage Subscription Related Information)</p>
5.	<p>The usage transaction determines the usage group(s) to use when calculating usage. These are retrieved from the usage subscription (with a fallback to the usage subscription type).</p> <p>This base package logic can be overridden by specifying an algorithm in the Usage Group Determination Override Algorithm field on the usage subscription type. If an algorithm is specified in this field, its logic overrides the existing method of determining the usage group.</p> <p>A base package algorithm (D2-DRVUSGRP) is provided that can be used here. This algorithm uses the CCB Rate Schedule extendable lookup used to map the rate schedules in Oracle Utilities Customer Care and Billing to usage groups. The algorithm's logic looks for a combination of rate history list entries on the usage subscription and the type of device configuration installed during the bill period to determine the usage group(s) to use for bill determinants calculation.</p>	
5.	<p>The usage transaction business object calculates usage based on the date range provided in the request.</p> <p>This processing is performed by an Enter algorithm on the “Calculate” state of the usage transaction business object.</p>	Enter Algorithm: D2-CALCUSG (Calculate Usage)
6.	<p>If the usage transaction has a sub usage transactions, it checks the status of each.</p> <p>This processing is performed by an Enter algorithm on the “Calculation in Progress” state of the usage transaction business object.</p>	Enter Algorithm: D2-CHKSUBUT (Check Sub Usage Transactions)

Step	Process	Objects
7.	<p>If the usage transaction is configured to require an approval, a To Do Entry is created.</p> <p>This processing is performed by an Enter algorithm on the “Approval in Progress” state of the usage transaction business object.</p>	<p>Enter Algorithm: D2-UTAP-TODO (Create Usage Transaction Requiring Approval To Do Entry)</p>
8.	<p>The usage transaction then sends the usage transaction to any subscribing systems.</p> <p>This processing is performed by an Enter algorithm on the “Sent” state of the usage transaction business object, and a processing method for the service provider for the subscribing systems.</p>	<p>Enter Algorithm: D2-SEND-USG (Send Usage)</p> <p>Processing Method: D2-HowToSendUSInformation (How To Send US Related Information)</p>

Integrating with Oracle Utilities Operational Device Management

This section provides an overview of how Oracle Utilities Meter Data Management supports integrations with Oracle Utilities Operational Device Management.

In an integration between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management:

- Oracle Utilities Meter Data Management is typically considered the “system of record” for service points (asset locations) and contacts, and for device installation information (MDM Install Events)
- Oracle Utilities Operational Device Management is typically considered the “system of record” for assets (devices)

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 Operational Device Management is based on data synchronization between the two systems.

Data Synchronization

The specific data synchronization flows supported between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management include the following:

- **Asset-Device Synchronization:** As new assets are created or changed in Oracle Utilities Operational Device Management, corresponding devices must be created or changed in Oracle Utilities Meter Data Management
- **Service Point/Contact – Asset Location/Contact:** As Service Points and/or Contacts are created or changed in Oracle Utilities Meter Data Management, corresponding Asset Locations and Contacts must be created or changed in Oracle Utilities Operational Device Management
- **Install Events– Asset Location/Disposition:** As devices are installed/removed in Oracle Utilities Meter Data Management, corresponding changes to an asset’s Disposition (location and status) must be made in Oracle Utilities Operational Device Management

This synchronization process is supported through a set of business objects, master configurations, batch controls, and pre-configured XAI Inbound Services. Refer to the *Oracle Utilities Integration for Device Operations Implementation Guide* for more information about this integration and the data synchronization processes used by this integration.

Inbound Data Synchronization Business Objects

The integration between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management uses the following inbound (to Oracle Utilities Meter Data Management) synchronization business objects (based on the F1-SYNCREQIN maintenance object):

Business Object	Description
D1-OngoingSyncRequestDevice	Device Ongoing Sync Request Used to synchronize devices in Oracle Utilities Meter Data Management based on assets created/updates in Oracle Utilities Operational Device Management

Outbound Data Synchronization Business Objects

The integration between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management uses the following outbound (from Oracle Utilities Meter Data Management) synchronization business objects (based on the F1-SYNC REQ maintenance object)

Business Object	Description
D1-ODMContactSyncRequest	ODM Sync Request Used to synchronize contacts in Oracle Utilities Operational Device Management based on contacts created/updated in Oracle Utilities Meter Data Management
D1-ODMInstallEventSyncRequest	ODM Install Event Sync Request Used to synchronize asset location/disposition in Oracle Utilities Operational Device Management based on install events created/updated in Oracle Utilities Meter Data Management
D1-ODMSPSyncRequest	ODM SP Sync Request Used to synchronize asset locations in Oracle Utilities Operational Device Management based on service points created/updated in Oracle Utilities Meter Data Management

Master Configurations

Master configurations are used to define aspects of the synchronization process, including resolution of foreign keys and the type of synchronization business objects to use for each type of data being synchronized.

The table below lists the master configurations used in data synchronization processing.

Master Configuration	Description
Master Data Synchronization Configuration	Lists all foreign key references that need resolution. Each one should reference the view that contains the external key / production key cross-reference. For entities that undergo both the initial and the ongoing sync, two views are specified. For entities that undergo the ongoing sync, an external system / ID type mapping is specified to cater for entities that might be synchronizing from more than one external system.

Master Configuration	Description
Seeder Sync Request Master Configuration	Lists the maintenance objects (device, device configuration, etc.) that require synchronization. Each references the synchronization business object that needs to be instantiated when processing a synchronization request for that maintenance object. For maintenance objects that undergo both initial and the ongoing synchronization, two business objects are specified.
ODM Integration Master Configuration	<p>Specifies the external system used to represent Oracle Utilities Operational Management, and the URL for the Oracle Utilities Operational Management application.</p> <p>Also specifies the outbound message types used to send synchronization requests to Oracle Utilities Operational Management.</p>

Integrating with Oracle Utilities Analytics

Oracle Utilities Meter Data Management can also be integrated with Oracle Utilities Analytics to allow users to view analytic data based on usage, events, and other data tracked in Oracle Utilities Meter Data Management.

Oracle Utilities Meter Data Management Analytics Products

Oracle Utilities Meter Data Management Analytics comprises the following products:

- **Oracle Utilities Meter Data Analytics:** dashboards, dashboard pages, and analytics used to view usage, event, and other data from Oracle Utilities Meter Data Management
- **Oracle Utilities Meter Data Management Extractors and Schema:** the star schema and extract programs used by Oracle Utilities Meter Data Analytics.

Oracle Utilities Meter Data Management Analytics Components

Oracle Utilities Meter Data Management includes the following components used when integrating with Oracle Utilities Analytics.

Master Configurations

Master configurations are used to configure the extract, transform, and load (ETL) process used with Oracle Utilities Analytics. See **Master Configurations** on page 2-4 for more information about the master configurations used with Oracle Utilities Analytics and Oracle Utilities Meter Data Analytics.

Batch Controls

Oracle Utilities Meter Data Management data extract and initial load is performed via a set of batch controls provided in the base package. See **Meter Data Management Business Intelligence Batch Controls** on page 14-10 for a list of batch controls used for extract and initial load of data for use with Oracle Utilities Analytics and Oracle Utilities Meter Data Analytics.

For More Information

Refer to the following documentation for more information about Oracle Utilities Meter Data Management Business Intelligence:

- Oracle Utilities Extractors and Schema for Oracle Utilities Meter Data Management Data Mapping Guide
- Oracle Utilities Analytics Dashboards for Meter Data Analytics Metric Reference Guide
- Oracle Utilities Analytics User's Guide

Integrating with Oracle Utilities Customer Self Service

Oracle Utilities Meter Data Management can be integrated with Oracle Utilities Customer Self Service to allow utilities to allow their customers to view their usage data, create self-service meter readings, and perform rate comparisons (via the Oracle Utilities Customer Care and Billing integration to Oracle Utilities Meter Data Management). This section includes:

- **Meter Data Management Services**
- **Configuring Rate Compare Usage Adjustment Profiles**

Meter Data Management Services

Oracle Utilities Meter Data Management provides the following services used by Oracle Utilities Customer Self Service.

- **Create Self-Service Meter Read:** Allows users to submit their own meter reads via the Oracle Utilities Self Service application
 - **Service Script:** WX-CrSSMRead
 - **XAI Inbound Service:** WX-CreateSelfServiceMeterRead
- **Get Scalar Consumption Summary:** Retrieves consumption data for display in the Oracle Utilities Self Service application
 - **Service Script:** WX-GetSCsum
 - **XAI Inbound Service:** WX-GetScalarConsumptionSummary
- **Get Usage Overview:** Retrieves an overview of a customer's usage for a user-specified duration for display in the Oracle Utilities Self Service application
 - **Service Script:** WX-GetUsgOVw
 - **XAI Inbound Service:** WX-GetUsageOverview
- ***Multiple Account TOU Usages by Service Request Type:** Uses the Get Usage Details service to retrieve usage for a list of accounts for display in the Oracle Utilities Self Service application. Each account's usage is summarized by service type, UOM and SQL.
 - **Service Script:** WX-MulAccTOU
 - **XAI Inbound Service:** WX-MultipleAccountTOUUsagesByServiceType
- ***Multiple Account Usages by Service Request Type:** Uses the Get Usage Details service to retrieve usage for a list of accounts for display in the Oracle Utilities Self Service application. Each account's usage is summarized by service type, UOM and SQL.
 - **Service Script:** WX-MulAccCmp
 - **XAI Inbound Service:** WX-MultipleAccountUsagesByServiceType
- ***Multiple Account Usage Download Request:** Uses the Get Usage Details service to retrieve usage for a list of accounts (by usage subscription) for display in the Oracle Utilities Self Service application.
 - **Service Script:** WX-MulAccUsg
 - **XAI Inbound Service:** WX-MultipleAccountUsagesDownload
- **Get Usage Details:** Retrieves usage details for a customer for a user-specific time period (year, month, day) for display in the Oracle Utilities Self Service application
 - **Business Service:** WX-RETWSSTOUMapping
 - **XAI Inbound Service:** WX-RETWSSTOUMappingService

- ***Usage Adjustment Retrieval:** Retrieves usage adjustment types based on a user-specified rate schedule. Used by the Rate Compare feature of the Oracle Utilities Self Service application.
 - **Service Script:** WX-RetUsgAdj
 - **XAI Inbound Service:** WX-RetUsgAdj
- ***Usage Adjustment Retrieval:** Retrieves usage adjustment types based on a user-specified rate schedule. Used by the Rate Compare feature of the Oracle Utilities Self Service application.
 - **Service Script:** WX-RetUsgAdj
 - **XAI Inbound Service:** WX-UsageAdjustmentRetrieval

These services are based on the service scripts and business services noted above, and are invoked via the corresponding XAI Inbound Services.

Refer to the *Oracle Utilities Customer Self Service Implementation Guide* for more information about integrating Oracle Utilities Meter Data Management with Oracle Utilities Customer Self Service.

*These services are available with Oracle Utilities Customer Self Service v2.1 and later.

Configuring Rate Compare Usage Adjustment Profiles

This section describes the steps involved in setting up usage adjustment profiles in Oracle Utilities Meter Data Management for use with the Rate Compare feature in Oracle Utilities Customer Self Service.

Overview

This section provides an overview of the steps involved in setting up usage adjustment profiles. The sections that follow provide additional details and examples for each type of data that must be configured.

1. Create a characteristic type and values for each type of usage adjustment to be made available for rate comparison purposes. For example, if you wanted to allow adjustments for installation of solar panels, use of an energy-efficient appliance, or use of an electric vehicle, you would create three characteristic types (one for each type of adjustment), and values for each option within each type (each type of appliance or electric vehicle supported).
2. Create profile (stand-alone) measuring components that correspond to each characteristic type value.
3. Create profile data for each measuring component. This is interval data that represents the impact of the usage adjustment on a customer's consumption.
4. Create a profile factor for each usage adjustment type.
5. Create factor values for each factor. These values correspond to the characteristic type value, and link characteristic type values to profile measuring components.
6. Create one (or more) Profile Accumulation usage rules that will apply the usage adjustment profile to the customer's interval consumption when calculating usage for the rate comparison request.
7. Create entries in the CCB Rate Schedule extendable lookup to associate the usage group that contains the "Profile Accumulation" usage rule to a rate schedule in Oracle Utilities Customer Care and Billing.
8. Create entries in the Usage Adjustment Types extendable lookup for each usage adjustment type.

9. Set up the Rate Compare Configuration section of the Self-Service Master Configuration to link usage adjustment factors and usage adjustment types to rate schedules Oracle Utilities Customer Care and Billing (defined in the CCB Rate Schedule extendable lookup).

Characteristic Types

Create a characteristic type and values for each type of usage adjustment you wish to make available to customer self-service users. For example, to create a usage adjustment profile for use of an electric vehicle, you would create an "electric vehicle" characteristic type and define values for each type of electric vehicle users can select.

Example: Electric Vehicles

- **Characteristic Type:** ELEC_VEH
- **Description:** Electric Vehicles
- **Type of Char Value:** Predefined Value
- **Characteristic Values:**

Characteristic Value	Description
LEAF	Nissan Leaf
TESLA	Tesla Model S
VOLT	Chevrolet Volt

Profile Measuring Components

Create profile measuring components for each characteristic type value. These measuring components will be used to store profile data for each type of usage adjustment.

Note: The base package does not include stand-alone measuring component/measuring component type business objects, but the demonstration database contains "Standalone Interval" and "Standalone Interval Measuring Component Type" business objects that can be used to create profile measuring components and types.

Measuring Component Type

Example: Profile Measuring Component Type

- **Measuring Component Type:** KWH-PROFILE
- **Description:** KWH Profile
- **Measuring Component Business Object:** Standalone Interval (demo)
- **Measurement Business Object:** Measurement
- **Service Type:** Electric
- **Allow Negative Consumption:** Allowed
- **Consumptive / Subtractive:** Consumptive
- **Seconds Per Interval:** 01:00:00
- **Value Identifiers:**
 - **Value Identifier Type:** Measurement
 - **Short-Hand Description:** kWh
 - **UOM:** Kilowatt-Hours

Measuring Components

Example: "Nissan Leaf" Profile Measuring Component

- **Measuring Component Type:** KWH Profile
- **Number of Digits Left:** 5
- **Number of Digits Right:** 5
- **Time Zone:** US Pacific Time
- **Status:** Active
- **How To Use:** Additive
- **External ID:** Nissan Leaf

Example: "Tesla Model S" Profile Measuring Component

- **Measuring Component Type:** KWH Profile
- **Number of Digits Left:** 5
- **Number of Digits Right:** 5
- **Time Zone:** US Pacific Time
- **Status:** Active
- **How To Use:** Additive
- **External ID:** Tesla Model S

Example: "Chevrolet Volt" Profile Measuring Component

- **Measuring Component Type:** KWH Profile
- **Number of Digits Left:** 5
- **Number of Digits Right:** 5
- **Time Zone:** US Pacific Time
- **Status:** Active
- **How To Use:** Additive
- **External ID:** Chevrolet Volt

Profile Data

Create profile data for each profile measuring component. This data represents the impact of the usage adjustment on a customer's consumption.

Note that this profile data can (and often will) include negative interval values to represent the difference in consumption applicable for the usage adjustment type. For example, if an energy-efficient electric clothes dryer uses an average of 30 kilowatt hours less per month than an average electric clothes dryer, profile data for that appliance might be a "straight line" hourly profile (a profile in which all intervals are of the same value) in which each value equals "0.042" (30 kWh per month divided by an average of 720 hours per month).

Factors / Factor Values

Create a profile factor for each usage adjustment type. This factor should use the "Factor Characteristic Source N/A Algorithm" to derive the appropriate characteristic type value based on the factor value, and should reference the characteristic type created earlier.

Factor

Example. Electric Vehicle Factor

- **Factor:** ELECTRIC_VEHICLE
- **Description:** Electric Vehicle
- **Factor Class:** Profile
- **Characteristic Source Algorithm:** Factor Characteristic Source N/A Algorithm
- **Factor Characteristic Type:** Electric Vehicle

Factor Values

Example: "Nissan Leaf" Factor Value

- **Factor:** Electric Vehicle
- **Factor Characteristic Type:** Electric Vehicle
- **Factor Characteristic Value:** LEAF
- **Effective Date/Time:** 01-01-2014 12:00:00AM
- **Profile:** Nissan Leaf, KWH Profile

Example: "Tesla Model S" Factor Value

- **Factor:** Electric Vehicle
- **Factor Characteristic Type:** Electric Vehicle
- **Factor Characteristic Value:** LEAF
- **Effective Date/Time:** 01-01-2014 12:00:00AM
- **Profile:** Tesla Model S, KWH - 60 Minutes

Example: "Chevrolet Volt" Factor Value

- **Factor:** Electric Vehicle
- **Factor Characteristic Type:** Electric Vehicle
- **Factor Characteristic Value:** LEAF
- **Effective Date/Time:** 01-01-2014 12:00:00AM
- **Profile:** Chevrolet Volt / KWH - 60 Minutes

Profile Accumulation Usage Group and Rule

Create a usage group that contains a "Profile Accumulation" usage rule. This rule will calculate usage by accumulating historical usage with profile data, based on a selected profile factor value.

Note: Profile Accumulation rules should use eligibility criteria to ensure they are only executed when the "Calculation Mode" on the usage transaction is set to "Hypothetical Calculation" (D2HC).

Example: Profile Accumulation Usage Rule

- **Usage Group:** Electric Residential Interval KWH
- **Usage Rule:** KWH_PROFILE_ACCUMULATION
- **Sequence:** 10

- **Description:** KWH Profile Accumulation
- **Category:** Usage Calculation
- **Vector Source Configuration:**
 - **Vector Type:** Channels Linked to Usage Subscription
 - **Unit of Measure:** Kilowatt-Hours
 - **Time of Use:**
 - **Service Quantity Identifier:**
 - **Target SPI:** 01:00:00
- **Result Processing Configuration:**
 - **Apply TOU Map to Derived Vector:** Yes
 - **TOU Map:** Summer / Winter, 15 minute interval
- **Result Storage Configuration:**
 - **Insert Primary SQ Entry:** Yes
 - **Save Derived Vector:** No
 - **Service Quantity Identifier:**
 - **Extract Interval Data:** No

CCB Rate Schedule Extendable Lookup

Create an entry in the "CCB Rate Schedule" extendable lookup to associate the usage group that contains the "Profile Accumulation" usage rule to an applicable rate schedule in Oracle Utilities Customer Care and Billing.

Example: CCB Rate Schedule extendable lookup

- **Rate:** E-INT-RES
- **Description:** Electric Residential Interval Rate
- **Default Usage Group:** Electric Residential Interval KWH (E-INT-RES)

Usage Adjustment Types Extendable Lookup

Create an entry in the "Usage Adjustment Type" extendable lookup for each type of usage adjustment that will be available to customer self-service users. These entries are used in the Rate Compare Configuration section of the Self-Service Master Configuration (see below).

Example: Electric Vehicle

- **Usage Adjustment Type:** ELEC_VEHICLE
- **Description:** Purchase of Electric Vehicle
- **Override Description:** Purchase of Electric Vehicle
- **External Reference ID:** Purchase of Electric Vehicle

Self-Service Master Configuration - Rate Compare Configuration

Configure the "Rate Compare Configuration" section of the Self-Service Master Configuration to associate usage adjustment factors and usage adjustment types with an applicable rate schedule in Oracle Utilities Customer Care and Billing (defined in the CCB Rate Schedule extendable lookup).

Example: Self-Service Master Configuration

- **Factor Characteristic Type Indicating No Value Variation:** N/A

- **External Reference Factor Value Characteristic Type:** External Reference ID
- **Minimum Days of Usage Adjustment Data:** 2
- **Rate / Usage Adjustments:**
 - **Rate:** Electric Residential - Interval KWH
 - **Usage Adjustments:**
 - **Usage Adjustment Factor:** Electric Vehicle
 - **Usage Adjustment Type:** Purchase of Electric Vehicle

Oracle Utilities DataConnect

Oracle Utilities DataConnect facilitates extraction of data from Oracle Utilities Meter Data Management for use in external applications such as analytics applications and energy management systems. This section describes how Oracle Utilities DataConnect works and how to implement and configure the system to support data extract processing. This includes the following:

- **DataConnect Data Extracts**
- **Consumption Extracts**
- **Master Data Extracts**
- **Extract Flat File Formats**

DataConnect Data Extracts

Oracle Utilities DataConnect can be used to extract two types of data:

- Usage measurement data for service points (based on the Service Point Types defined for a Consumption Extract Type (see below)

When extracting usage measurement data, Oracle Utilities DataConnect extracts both interval measurement data as well as frequently read scalar usage data (which is converted into interval measurements during the extraction process).

- Individual service points and install events (based on Service Point Types defined in an extract algorithm)

The extraction processes used for each type of data are separate, but extracted data can be correlated in external systems through service point data included in each type of extract. All data extracts contain the following data elements that can be used for this correlation:

- **Service Point ID:** The service point ID in Oracle Utilities Meter Data Management
- **CIS Service Point ID:** The ID used for the service in a customer information system (CIS) such as Oracle Utilities Customer Care and Billing

External systems receiving data extracted from Oracle Utilities DataConnect can use either (or both) of these IDs to associate extract measurement data to extracted master data.

Consumption Extracts

When sending interval measurement usage data to an external system, both historical and current data needs to be extracted. Historical data can be extracted as part of an initial load process, and only needs to be provided during initial setup of the integration. Historical data should include data history for all active service points for a specified historical period. Current data should be extracted on a regular ongoing (or incremental) basis. However, in addition to sending current data, any historical corrections received by the system should be extracted as well.

Types of Extracts

There several types of consumption extracts:

- **Initial Load:** Initial load extract requests are created and submitted manually via the Consumption Extract Request portal. Consumption extract requests are based on a specified Consumption Extract Type (see below) and extraction date range. An initial load request must be created and submitted for each consumption extract type defined in the system.
- **Incremental / Ongoing (Current Data):** Incremental / ongoing extract requests can be manually created, but more often will be created via a batch process. The "Create Daily Consumption Extract Requests" batch control scans active consumption extract types and

creates a request for each one that has Frequency of "Automated Daily." Ad-hoc incremental requests can be created and submitted manually if needed.

- **Historical Correction:** Historical correction extracts are created via batch process. Algorithms on the Finalized state of the initial measurement and measurement business objects determine if a finalized initial measurement or rederived values are historical corrections. These algorithms create records which are evaluated by a batch process which extracts the measurements for the related initial measurements. See **Detecting Historical Corrections** on page 15-26 for more information about how historical corrections are detected and processed.

Consumption extract requests use the following types of administrative data. Refer to the *Oracle Utilities Meter Data Management User's Guide* for more information.

Consumption Extract Type

Consumption extract types define the specific parameters used when processing a consumption extract request. Consumption Extract Types control the service point type, type of measurement, and how the measurements are grouped into TOU periods if applicable. Consumption extract types also define the algorithm and batch processes to use when extracting data for different types of requests (initial load, interim, and historical).

There are two consumption extract type business objects provided with the base package:

- **Consumption Extract Type** (D2-ConsumptionExtractType): This business object retrieves interval data and converts it to a specified Target UOM and Interval Size. This business object does not support TOU mapping.
- **Consumption Extract Type with TOU Mapping** (D2-ConsumptionExtractTypeTOU): This business object retrieves interval data, converts it to a specified Target UOM and Interval Size, and maps it to a specific TOU map prior to extraction.

Consumption Extract Request Type

Consumption extract request types define informational details about specific types of consumption extract requests. A single business object is provided in the base package for consumption request types:

- **Consumption Request Type for DataConnect** (D2-IntervalDataExtRepoType)

Historical vs. Current Data

The "Extract Through Date/Time" field on the Consumption Extract Type is used to differentiate between current data (the most recently extracted data) and historical corrections, and is set to the last date on which data was extracted for that extract type. For example, if data is extracted on June 1, 2015, the "Extract Through Date/Time" would be set to "June 1, 2015 12:00AM." If/when data is extracted the next day, "Extract Through Date/Time" would be updated to "June 2, 2015 12:00AM."

When evaluating data for extract:

- Interval data is considered current if its measurement date time is after the "Extract Through Date/Time".
- Interval data is considered a historical correction if its measurement date time is on or before the "Extract Through Date/Time".

Detecting Historical Corrections

Historical data changes to an initial measurement can be detected when it enters the Finalized state. If the initial measurement is determined to be for a historical period by comparing its end date/time against the "Extracted Through Date/Time" on the Consumption Extract Type, a general process record will be written for the initial measurement so that measurements for it can be extracted. In addition, rederived values on final measurements can also trigger the creation of a general process record for related initial measurements.

The following algorithms are used in this process:

- The "Create General Process Record if IMD is Historical Correction" algorithm is used to determine if a finalized initial is a historical correction. If it is, the algorithm creates a general process record for the initial. This algorithm is provided in the base package, but not specified on initial measurement business objects by default. This algorithm should be defined as an Enter algorithm on the Finalized state of the initial measurement business objects.
- The "Create General Process Record for Re-derived Values" algorithm creates general process records for initial measurements associated with rederived values. Processing will proceed as if a historical correction came in through an initial measurement. This algorithm is provided in the base package, but not specified on measurement business object by default. This algorithm should be defined as an Enter algorithm on the Re-derive state of the final measurement business object.

Consumption Extract Requests

Initial load and ongoing consumption extracts are created via consumption extract requests. While extracts of these types can be created via adhoc submission of a batch job, requests are the preferred method for these types of consumption extracts.

The consumption extract request business object lifecycle includes logic that maintains and updates the "Extraction Through Date/Time" field on Consumption Extract Types, which is used to determine if daily requests should be created by the "Create Daily Consumption Extract Requests" batch control, and detect historical corrections.

See **Understanding Request Processing** on page 15-28 for more information about how consumption extract requests are processed.

Request Business Object

A single business object is provided in the base package for consumption extract requests:

- **Consumption Request for DataConnect (D2-IntervalDataExtRepository):** This business object contains the information and lifecycle responsible for submitting the extract job, monitoring the run until it's finished, and then updating the Consumption Extract Type's "Extract Through Date/Time" on the Consumption Extract Type. This business object is based on the Request (F1-REQ) maintenance object.

Algorithms

The consumption extract process uses a set of base package algorithms to extract and format the interval data for export. These algorithms are specified in the "Algorithms" section on the Consumption Extract Type as appropriate.

These algorithms can be configured to allow for extraction of data for frequently-read scalar measuring components as well as interval measuring components. Frequently-read scalar measuring components are defined as scalar measuring components whose Read Method is set to "Automatic Read." When extracting measurements for frequently-read scalar measuring components, scalar measurements are converted to interval measurements as part of the extraction process. This conversion uses the profile associated with the measuring component type. If not profile can be found, the interval data uses a flat line profile.

Initial Load / Incremental / Ongoing Requests:

The following algorithms are used to extract and format interval data for initial load and incremental / ongoing requests:

- **Extract Initial Load/Ongoing Consumption for DataConnect (D2-IDEXTPRD):** This algorithm retrieves a service point's consumption for a given period and writes the results to a flat file.

- **Extract Initial/Ongoing Consumption and Apply TOU Map for DataConnect (D2-IDEXTPTOU):** This algorithm retrieves a service point's consumption for a given period, applies a TOU Map to the consumption, and writes the results to a flat file.

Historical Corrections:

The following algorithms are used to extract and format interval data for historical correction requests:

- **Extract Historical Correction Consumption for DataConnect (D2-IDEXTIMD):** This algorithm retrieves historical correction consumption for a service point and writes the results to a flat file.
- **Extract Historical Corrections and Apply TOU Map for DataConnect (D2-IDEXTITOU):** This algorithm retrieves historical correction consumption for a service point, applies a TOU map to the consumption, and writes the results to a flat file.

The batch control that executes these algorithms specify the path and file name for output files. Use the Algorithm portal for more information about these algorithms.

Batch Controls

The consumption extraction process uses a set of base package batch controls to extract and format the interval data for export.

- **Create Daily Consumption Extract Requests (D2-CRERQ):** This batch process scans for active Consumption Extract Types, and for each one that has Frequency of Automated Daily creates a pending request (see **Understanding Request Processing** on page 15-28 for more information). This process should be scheduled to run daily (or at another regular interval).

The following sample batch controls are provided to extract and format interval data. Unique batch controls of each of these is required for each consumption extract type. You should create custom versions of the above batch controls for each consumption extract type in your implementation. Extract type-specific versions of these batch controls should be specified in the "Batch Control" section on the Consumption Extract Type as appropriate.

- **Initial Load/Ongoing Consumption Extract (D2-IDEPD):** This batch process extracts interval data for a specified period. This batch control uses the "Initial Load/Ongoing Extract Algorithm" defined on the Consumption Extract Type.
- **Historical Corrections Consumption Extract (D2-IDEHC):** This batch process extracts interval data for historical corrections. This batch control uses the "Historical Corrections Extract Algorithm" defined on the Consumption Extract Type.

Use the Batch Control portal for more information about these batch controls. The extract batch controls contain parameters that can be used to specify details (including path and file name for this file.) for a delimited flat file containing extracted data.

Understanding Request Processing

This section outlines how consumption extract requests are created and processed.

Step	Process	Base Package Meter Data Management Objects
1.	<p>A consumption extract request is created either via batch or manually.</p> <p>The Create Daily Consumption Extract Requests (D2-CRERQ) batch control is used to create ongoing extract requests.</p> <p>Requests begin in the Pending state.</p>	

Step	Process	Base Package Meter Data Management Objects
2.	If the request is discarded by a user, request transitions into the Discarded state.	
3.	<p>When a request enters the Submit Job state:</p> <ul style="list-style-type: none"> An Enter algorithm sets the extract batch control (from the Consumption Extract Type) type to Timed. A Timed batch process will be automatically initialized on a regular basis. An enter algorithm creates a batch job based on the Consumption Extract Type and the requested extract period. Fallback parameters are used if the corresponding batch parameters are not configured on the batch control. 	<p>Enter Algorithm: Turn On Timer of Wait for Run to Finish Monitor Batch Job (D2-TURNONTM)</p> <p>Enter Algorithm: Submit the Extract Batch Job (D2-SUBMITJOB)</p>
4.	The request then transitions to the Wait for Run to Finish state. An enter algorithm monitors the request batch job. The request remains in that state until the extract batch job is finished.	Enter Algorithm: Wait for Consumption Extract Request's Batch Run to Finish (D2-WTRUNFIN)
	The algorithms specified in the Algorithms section of the Consumption Extract Type are executed by the batch process to perform the extract process.	<p>Initial Load / Ongoing Requests:</p> <ul style="list-style-type: none"> Extract Initial Load/Ongoing Consumption for DataConnect (D2-IDEXTPRD) Extract Initial/Ongoing Consumption and Apply TOU Map for DataConnect (D2-IDEXTPTOU) <p>Historical Corrections:</p> <ul style="list-style-type: none"> Extract Historical Correction Consumption for DataConnect (D2-IDEXTIMD) Extract Historical Corrections and Apply TOU Map for DataConnect (D2-IDEXTITOU)
6.	<p>If an error occurs, the request transitions into the Error state.</p> <ul style="list-style-type: none"> An Enter algorithm resets the batch control's type to Not Timed (which was set to Timed on the "Wait for Run to Finish" state). <p>A user can manually resubmit the request, which returns it to the Submit Job state (see step 3 above).</p>	Enter Algorithm: Turn Off Timer of Wait for Run to Finish Monitor Batch Job (D2-TURNOFFTM)

Step	Process	Base Package Meter Data Management Objects
7.	<p>When the request is complete, it transitions to the Complete state.</p> <ul style="list-style-type: none"> An enter algorithm updates the "Extracted Through Date/Time" on the consumption extract type. An Enter algorithm resets the batch control's type to Not Timed (which was set to Timed on the "Wait for Run to Finish" state). 	<p>Enter Algorithm: Update Extraction Through Date/Time (D2-UPDCET)</p> <p>Enter Algorithm: Turn Off Timer of Wait for Run to Finish Monitor Batch Job (D2-TURNOFFTM)</p>

Implementing Oracle Utilities DataConnect for Consumption Extracts

Setting up consumption extracts involves the following steps:

1. Create Consumption Request Types

You should create a consumption request type for each unique type of request required by your implementation.

See **About Consumption Request Types** and **Defining Consumption Request Types** in the *Oracle Utilities Meter Data Management User's Guide* for more information.

2. Create Consumption Extract Types

You should create a consumption extract type for each unique combination of output details (target UOM, target interval size, TOU map/template) and measurement selection criteria,

See **About Consumption Extract Request Types** and **Defining Consumption Extract Request Types** in the *Oracle Utilities Meter Data Management User's Guide* for more information.

- Add the "Create General Process Record if IMD is Historical Correction" historical correction algorithm as an Enter algorithm on the Finalized state of the initial measurement business objects.
- Add the "Create General Process Record for Re-derived Values" historical correction algorithm as an Enter algorithm on the Finalized state on the measurement business object.
- Create and submit initial load consumption extract requests for each consumption extract type you created earlier.
- Set up batch processes for daily extract requests and historical corrections.

Batch processing for consumption extracts should include the following:

- The **Create Daily Consumption Extract Requests** (D2-CRERQ) batch control should be configured to run on a regular (i.e. daily) schedule to create ongoing consumption extract requests.
- The **Request Monitor (Deferred)** (F1-SUBRQ) batch control should be used to monitor for pending requests and transition them to the "Submit Job" state.
- Historical corrections consumption extract batch controls (based on D2-IDEHC, one for each consumption extract type), should be configured to run on a regular basis to check for and create historical correction extracts.

Master Data Extracts

Extraction of service point and install event master data is performed through use of data synchronization, audit algorithms, business objects and related batch processes.

Batch processes are used to create initial load extracts for service point and install event master data. Following the initial load, data synchronization requests are created when master data is changed.

Maintenance Object Audit Algorithms

Detecting changes in service point and install event master data can be detected audit algorithms on the Service Point (D1-SP) and Instal Event (D1-INSTLEVT) maintenance objects.

The Generic Change Data Capture (F1-GCHG-CDCP) algorithm type provided in the base package can be used to create audit algorithms for the Service Point and Install Event maintenance objects.

In addition, the "Device Change Data Capture (Install Event-Based)" (D1-IEDV-CDCP) algorithm type provided in the base package can be used to create an audit algorithm on the Device maintenance object to determine if an install event sync request record is to be created. A change in a related device's details will instantiate an Install Event Extract Sync Request (of the type defined for the "Install Event Sync Request BO" algorithm parameter) if one does not already exist in the initial state for the Install Event. The base package contains an algorithm based (D1-IEDV-CDCP) on this algorithm type.

Business Objects and Algorithms

The maintenance object audit algorithms create data synchronization requests based on the "Sync Request BO" maintenance object options. Extraction of service points and install events are supported by the following data synchronization business objects.

- **SP Sync for DataConnect** (D1-ExternalRepositorySPSync): used to extract service point information. This business objects should be defined as a value for the "Sync Request BO" options on the Service Point maintenance object.
- **Install Event Sync for DataConnect** (D1-ExternalRepositoryIESync): used to extract install event information. This business objects should be defined as a value for the "Sync Request BO" options on the Install Event maintenance object.

These business objects use the following Pre-Processing algorithms to take initial data snapshots, and define the batch control used to extract data and export to a flat file:

- **Capture SP Initial Snapshot for DataConnect** (D1-SPEINISNP)
- **Capture Install Event Initial Snapshot for DataConnect** (D1-IEEINISNP)

These algorithms specify the batch control used for the extract process (see below).

The Sync Request Monitor batch control (F1-SYNRQ) monitors synchronization requests in the Pending state and executes Monitor algorithms that check for related synchronization requests, and transitions them to the "Determine if Sync Needed" state.

Enter algorithms on the "Determine If Sync Needed" states extract a final snapshot of the data to extract and export.

- **Capture SP Final Snapshot for DataConnect** (D1-SPEFNISNP)
- **Capture Install Event Final Snapshot for DataConnect** (D1-IEEFNISNP)

The "Prepare Delimited Extract Data" Enter algorithm (D1-PRPEXTDTA) on the "Send Request" state prepares the data for extraction, and creates a general process record for the synchronization request (based on the batch control defined by the pre-processing algorithm). Note that this algorithm is defined on the "Send Request" state if the Generic Sync for DataConnect (D1-ParentExternalRepositorySyn) parent business object.

Batch Controls

The master data extraction process uses a set of batch processes to create data synchronization requests and extract files.

The following batch processes are used to create initial synchronization requests:

- **SP Initial Load for DataConnect** (D1-SPEIL): This batch control creates initial synchronization requests for service points.
- **Install Event Initial Load for DataConnect** (D1-IEEIL): This batch control creates initial synchronization requests for install events.

These batch processes should be run to create initial load data synchronization requests based on the "Sync for DataConnect" business objects.

The following batch processes are used to create extract files from synchronization requests.

- **SP Extract for DataConnect** (D1-SPESR): This batch control creates extract file(s) that contain service point information.
- **Install Event Extract for DataConnect** (D1-IEESR): This batch control creates extract file(s) that contain install event information.

These batch controls are defined as values for the "Batch Control for Extract" algorithm parameters on the Pre-Processing algorithms on the "Sync for DataConnect" business objects (see above).

Use the Batch Control portal for more information about these batch controls. The extract batch controls contain parameters that can be used to specify details (including path and file name for this file.) for a delimited flat file containing extracted data.

Implementing Oracle Utilities DataConnect for Master Data Extracts

Setting up master data extracts involves the following steps:

1. Add audit algorithms on the Service Point and Install Event maintenance objects.
2. Add the DataConnect synchronization request business objects as "Sync Request BO" options on the Service Point and Install Event maintenance objects.
3. Add the "Install Event's Device Change Data Capture" algorithm as an audit algorithm on the Device maintenance object.
4. Execute initial load batch processes for service points and install events.
5. Once the initial load synchronization has been executed, changes to service points, install events, or related devices will trigger the creation of new synchronization requests and resulting extraction files.

Extract Flat File Formats

This section outlines details concerning the format of output flat files created by consumption and master data extracts.

Data Areas

The data included in each file is based on a specific data area. For each data area, this section provides a table that lists the name and corresponding schema element and metadata field for each data element extracted by the data area.

Consumption Extract Snapshot

The Consumption Extract Snapshot data area is used for interval consumption extract.

Data Area: D2-IntervalDataExtRepoSnapshot

Data Element	Schema Element	Metadata Field
CIS Service Agreement ID	cisSAId	CIS_EXT_SA_ID
Service Point ID	spId	D1_SP_ID
CIS Service Point ID	cisSPId	CIS_EXT_SP_ID
Unit of Measure	uom	D1_UOM_CD
Service Quantity Identifier	sqi	SVC_QTY_IDNTFR_CD
Seconds Per Interval	secondsPerInterval	SECONDS_PER_INTERVAL
Measurement Value	measurementValue	MSRMT_VAL
Measurement Date	measurementDate	MSRMT_DATE
Measurement Time	measurementTime	MSRMT_TIME
Estimate Indicator	estimateIndicator	ESTIMATION_IND_FLG
Estimate Indicator Description	estimateIndicatorDescription	DESCR100

Consumption Extract Snapshot (TOU Mapping)

The Consumption Extract Snapshot (TOU Mapping) data area is used for interval consumption extract mapped to TOU periods.

Data Area: D2-IntervalDataExtRepoTOUSnap

Data Element	Schema Element	Metadata Field
CIS Service Agreement ID	cisSAId	CIS_EXT_SA_ID
Service Point ID	spId	D1_SP_ID
CIS Service Point ID	cisSPId	CIS_EXT_SP_ID
Unit of Measure	uom	D1_UOM_CD
Time of Use Period	tou	D1_TOU_CD
Service Quantity Identifier	sqi	SVC_QTY_IDNTFR_CD
Seconds Per Interval	secondsPerInterva	SECONDS_PER_INTERVAL
Measurement Value	measurementValue	MSRMT_VAL
Start Date	startDate	START_DT
Start Time	startTime	START_TM

Data Element	Schema Element	Metadata Field
End Date	endDate	END_DT
End Time	endTime	D2_END_TIME

Service Point Snapshot for DataConnect

The Service Point Snapshot for DataConnect data area is used for service point master data extract.

Data Area: D1-ExternalRepositorySPSnapsht

Data Element	Schema Element	Metadata Field
Service Point ID	spId	D1_SP_ID
CIS Service Point ID	cisSpId	CIS_EXT_SP_ID
Address	address1	DESCR254
City	city	CITY
State	state	STATE
Postal Code	postal	POSTAL
Service Point Type Code	spType	D1_SP_TYPE_CD
Service Point Type Description	spTypeDescription	DESCR100
Service Type Code	serviceType	D1_SVC_TYPE_CD
Service Type Description	serviceTypeDescription	DESCR100
Market Code	market	MKT_CD
Market Description	marketDescription	DESCR100
Service Point Status Code	boStatus	BO_STATUS_CD
Service Point Status Description	boStatusDescription	DESCR100
Disconnect Location Flag	disconnectLocation	DISCONN_LOC_FLG
Disconnect Location Description	disconnectLocationDescription	DESCR100
Measurement Cycle Code	measurementCycle	MSRMT_CYC_CD
Measurement Cycle Description	measurementCycleDescription	DESCR100
Measurement Cycle Route Code	measurementCycleRoute	MSRMT_CYC_RTE_CD
Measurement Cycle Route Description	measurementCycleRouteDescription	DESCR100
Geocode Latitude	geocodeLatitude	D1_GEO_LAT
Geocode Longitude	geocodeLongitude	D1_GEO_LONG
Facility Level 1	facilityLevel1	FACILITY_LEVEL1
Facility Level 2	facilityLevel2	FACILITY_LEVEL2
Facility Level 13	facilityLevel3	FACILITY_LEVEL3

Install Event Snapshot for DataConnect

The Install Event Snapshot for DataConnect data area is used for install event master data extract.

Data Area: D1-ExternalRepositoryIESnapshst

Data Element	Schema Element	Metadata Field
Service Point ID	spId	D1_SP_ID
CIS Service Point ID	cisSpId	CIS_EXT_SP_ID
Meter ID	meterId	D1_DEVICE_ID
Device Type Code	deviceType	DEVICE_TYPE_CD
Device Type Description	deviceTypeDescription	DESCR100
Device Status Code	deviceBoStatus	BO_STATUS_CD
Device Status Description	deviceBoStatusDescription	DESCR100
Manufacturer Code	manufacturer	MANUFACTURER_CD
Manufacturer Description	manufacturerDescription	DESCR100
Model Code	model	D1_MODEL_CD
Model Description	modelDescription	DESCR100
Head-End System Code	headEndSystem	D1_SPR_CD
Head-End System Description	headEndSystemDescription	DESCR100
Badge Number	badgeNumber	D1_BADGE_NBR
Serial Number	serialNumber	D1_SERIAL_NBR
CIS Meter ID	cisMeterId	CIS_EXT_DEVICE_ID
Install Date/Time	installDateTime	D1_INSTALL_DT_TM
Removal Date/Time	removalDateTime	D1_REMOVAL_DT_TM

File Details

Specifics for how the flat files are created are defined on batch controls and algorithms used by the extract process.

- **File Name and Path:** Parameters on batch controls define the file name, path, and other details about the output file.
- **File Size and Contents:** Batch controls for initial load/ongoing consumption requests include a parameter to specify the number of service points to be included in each file.
- **Character Encoding and File Delimiter:** Parameters on the Prepare Delimited Extract Data (D1-PRPEXTDTA) algorithm are used to specify the character encoding and delimiter used in the flat files for master data extracts.

Example Extract File

The following example illustrates comma-separated interval data extracts based on the Consumption Extract Snapshot data area:

```
19502793-60E-KMUS,714532246966,19502793-60E-KMSP,KWH, ,  
3,600,1.366,2015-01-01,07.00.00,No  
19502793-60E-KMUS,714532246966,19502793-60E-KMSP,KWH, ,  
3,600,1.366,2015-01-01,08.00.00,No  
19502793-60E-KMUS,714532246966,19502793-60E-KMSP,KWH, .  
3,600,1.366.2015-01-01,09.00.00,No  
19502793-60E-KMUS,714532246966,19502793-60E-KMSP,KWH, ,  
3,600,1.366,2015-01-0,10.00.00,No  
19502793-60E-KMUS,714532246966,19502793-60E-KMSP,KWH, ,  
3,600,1.366,2015-01-01,11.00.00,No  
19502793-60E-KMUS,714532246966,19502793-60E-KMSP,KWH, ,  
3,600,1.366,2015-01-01,12.00.00,No  
19502793-60E-KMUS,71453224696,19502793-60E-KMSP,KWH, ,  
3,600,1.366,2015-01-01,13.00.00,No
```

Initial Measurement and Device Event XML Formats

This section provides details concerning the XML formats used when importing initial measurements and device events, including:

- **Initial Measurement Data XML Format**
- **Device Event XML Format**

Initial Measurement Data XML Format

This section describes the XML format used for inbound initial measurement data. This includes interval and scalar examples, descriptions of the individual XML elements, and the initial measurement data XML schema based on the D1-IMDSeeder business object.

Example - Interval Initial Measurement Data

```
<IMD-IMPORT>
  <serviceProvider>HEADEND-1</serviceProvider>
  <serviceProviderExternalId>MDCS-1</serviceProviderExternalId>
  <preVEE>
    </simdId>
    <dvcIdN>037090184721</dvcIdN>
    <mcId>135914144111</mcId>
    <mcIdN>123</mcIdN>
    <externalId>IMD1234567</externalId>
    <uom>KWH</uom>
    <stDt>2009-01-02-00.00.00</stDt>
    </stQty>
    <enDt>2009-01-03-00.00.00</enDt>
    </enQty>
    <imdType>D1IL</imdType>
    <mcIS>D1IN</mcIS>
    <inShift>N</inShift>
    <mcm>1.0</mcm>
  </nd>
  <tz>USPACIFIC</tz>
</externalTimeZone>
<spi>3600</spi>
</ccond>
<sts>
  <stsL>
    <s>1</s>
    <st>REGULAR</st>
  </stsL>
</sts>
<msrs>
  <mL>
    <s>1</s>
    <q>1.6</q>
    <sts>
      <stsL>
        <s>1</s>
        <st>REGULAR</st>
      </stsL>
    </sts>
  </mL>
  <mL>
    <s>2</s>
    <q>1.57</q>
    <sts>
      <stsL>
        <s>1</s>
        <st>REGULAR</st>
      </stsL>
    </sts>
  </mL>
</msrs>
```

```
        </mL>
        <mL>
          <s>3</s>
          <q>0.0</q>
          <sts>
            <stsL>
              <s>1</s>
              <st>MISSING</st>
              <s>2</s>
              <st>OUTAGE</st>
            </stsL>
          </sts>
        </mL>
        <mL>
          <s>4</s>
          <q>0.0</q>
          <sts>
            <stsL>
              <s>1</s>
              <st>MISSING</st>
              <s>2</s>
              <st>OUTAGE</st>
            </stsL>
          </sts>
        </mL>
        <mL>
          <s>5</s>
          <q>1.0</q>
          <sts>
            <stsL>
              <s>1</s>
              <st>REGULAR</st>
            </stsL>
          </sts>
        </mL>
        <mL>
          <s>6</s>
          <q>1.45</q>
          <sts>
            <stsL>
              <s>1</s>
              <st>REGULAR</st>
            </stsL>
          </sts>
        </mL>
        ...
      </msrs>
    </preVEE>
    <processData>
      </isShiftStartEnd>
      </isShiftedIntervals>
      </isTimeZoneConverted>
      </isErrorEncountered>
      </servicePointId>
      </installationConstant>
      </deviceId>
    </logs>
  </processData>
</IMD-IMPORT>
```

Example - Scalar Initial Measurement Data

```
<IMD-IMPORT>
  <serviceProvider>HEADEND-2</serviceProvider>
  <serviceProviderExternalId>MDCS-2</serviceProviderExternalId>
  <preVEE>
    <dvcIdN>037090184721</dvcIdN>
    <mcId>327604570580</mcId>
```

```

    <mcIdN>123</mcIdN>
    <externalId>IMD7654321</externalId>
    <uom>KWH</uom>
    <stDt>2009-01-31-11.25.00</stDt>
    </stQty>
    <enDt>2009-02-28-13.13.00</enDt>
    <enQty>110.00</enQty>
    <imdType>D1IL</imdType>
    <mcIS>D1SC</mcIS>
    <inShift>N</inShift>
    <mcm>1.0</mcm>
    <nd>5</nd>
    <tz>USPACIFIC</tz>
  </externalTimeZone>
</ccond>
<sts>
  <stsL>
    <s>1</s>
    <st>REGULAR</st>
  </stsL>
</sts>
</preVEE>
<processData>
  </isShiftStartEnd>
  </isShiftedIntervals>
  </isTimeZoneConverted>
  </isErrorEncountered>
  </servicePointId>
  </installationConstant>
  </deviceId>
  </logs>
  <deviceEventTypes>
    <deviceEventTypesList>
      <deviceEventType>BROKEN</deviceEventType>
    </deviceEventTypesList>
  </deviceEventTypes>
</processData>
</IMD-IMPORT>

```

Element Descriptions - Initial Measurement Data

The table below provides descriptions of the elements used in the initial measurement data XML format.

Element	Description
<{SERVICE_NAME}>	Root element containing an initial measurement. This element should match the name of the inbound service used to import the usage.
<serviceProvider>	Name of the head-end system (defined as a service provider) in MDM
<serviceProviderExternalId>	External ID of the head-end system.
<preVEE>	Element containing Pre VEE measurement data
<simdId>	The ID of a specific initial measurement

Element	Description
<dvcIdN>	Device Identifier Number, e.g. a Serial Number. The "type" of device identifier that the head-end system understands is configured within MDM - and so can differ per head-end.
<mcId>	Measuring Component ID
<mcIdN>	Measuring Component Identifier Number. Populated with channel ID. The "type" of measuring component identifier that the head-end system understands is configured within MDM
<externalId>	File name of ID of the XML document containing the measurement data
<uom>	Unit of Measure (Optional)
<stDt>	Start Date/Time. Required for interval measurement data. Optional for scalar measurement data. Must be in the following format: YYYY-MM-DD-HH.MM.SS Example:2008-12-31-00.30.00
<stQty>	Start Reading. Optional.
<enDt>	End Date/Time. Required. Must be in the following format: YYYY-MM-DD-HH.MM.SS Example:2008-12-31-00.30.00
<enQty>	End Reading. Required for scalar measurements.
<imdType>	Initial measurement data type. Valid values include: <ul style="list-style-type: none"> • D1IL (Initial Load) • D1MO (Manual) • D1ES (Estimation) <p>Defaults to D1IL (Initial Load) if not supplied.</p>
<mcIS>	Interval Scalar flag. Indicates if the data is for an interval or scalar measuring component.
<inShift>	Incoming Data Shift flag. Indicates if the device is DST aware.
<mcm>	Meter multiplier.
<nd>	Number of Dials
<tz>	Time Zone

Element	Description
</externalTimeZone>	External Time Zone
<spi>	Seconds-per-interval
<ccond>	Measurement condition
<sts>	Element containing status code information for entire measurement.
<stsL>	List of head-end system status codes for the entire measurement
<s>	Sequence
<st>	Head-end status codes applicable to the entire set of data.
<msrs>	Element containing measurement data
<mL>	Element containing an individual interval measurements. Used for interval measurement data only.
<s>	Sequence of the interval measurement
<dt>	Date and time of the interval measurement. Optional.
<q>	Quantity of the interval measurement
<ue>	Used-Edited flag. Indicates if the interval measurement has been manually edited
<fc>	Final Condition code for the interval measurement. Optional.
<sts>	Element containing lists of status codes for each interval measurement
<stsL>	Element containing a sequence/status pairing for each interval measurement
<s>	Sequence of the status code for this interval measurement.
<st>	Head-end status code of the interval measurement.
<processData>	Element containing processing-related data.
</deviceEventTypes>	Element containing one or more lists of device event types (used to capture Reader Remarks for scalar measurements)
<deviceEventTypesList>	Element containing one or more device event types (used to capture Reader Remarks for scalar measurements)

Element	Description
<deviceEventType>	A device event type code that represents a Reader Remark (used with scalar measurements)

Schema - IMD Seeder (D1-IMDSeeder) Business Object

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ouaf="http://
ouaf.oracle.com/" targetNamespace="http://oracle.com/D1-InitialLoadIMD.xsd"
elementFormDefault="qualified">
  <xsd:import namespace="http://ouaf.oracle.com/" />
  <xsd:element name="D1-InitialLoadIMD">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="initialMeasurementDataId" type="xsd:string"
minOccurs="0" />
        <xsd:element name="preVEE" minOccurs="0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="simdId" type="xsd:string" minOccurs="0" />
              <xsd:element name="dvcIdN" type="xsd:string" minOccurs="0" />
              <xsd:element name="mcId" type="xsd:string" minOccurs="0" />
              <xsd:element name="mcIdN" type="xsd:string" minOccurs="0" />
              <xsd:element name="externalId" type="xsd:string" minOccurs="0" /
            >
            <xsd:element name="uom" type="xsd:string" minOccurs="0" />
            <xsd:element name="externalUOM" type="xsd:string" minOccurs="0"
          />
          <xsd:element name="stDt" type="xsd:dateTime" minOccurs="0" />
          <xsd:element name="stQty" type="xsd:decimal" minOccurs="0" />
          <xsd:element name="enDt" type="xsd:dateTime" minOccurs="0" />
          <xsd:element name="enQty" type="xsd:decimal" minOccurs="0" />
          <xsd:element name="imdType" minOccurs="0">
            <xsd:simpleType>
              <xsd:restriction base="xsd:string">
                <xsd:enumeration value="estimation" />
                <xsd:enumeration value="imdSeeder" />
                <xsd:enumeration value="initialLoad" />
                <xsd:enumeration value="manual" />
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="inShift" minOccurs="0">
            <xsd:simpleType>
              <xsd:restriction base="xsd:string">
                <xsd:enumeration value="notShifted" />
                <xsd:enumeration value="shifted" />
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="mcm" type="xsd:decimal" minOccurs="0" />
          <xsd:element name="nd" type="xsd:decimal" minOccurs="0" />
          <xsd:element name="tz" type="xsd:string" minOccurs="0" />
          <xsd:element name="externalTimzone" type="xsd:string"
minOccurs="0" />
          <xsd:element name="spi" type="xsd:int" minOccurs="0" />
          <xsd:element name="ccond" minOccurs="0">
            <xsd:simpleType>
              <xsd:restriction base="xsd:string">
                <xsd:enumeration value="301000" />
                <xsd:enumeration value="901000" />
                <xsd:enumeration value="501000" />
                <xsd:enumeration value="101000" />
                <xsd:enumeration value="100000" />
                <xsd:enumeration value="201000" />
                <xsd:enumeration value="401000" />
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:schema>

```

```

        <xsd:enumeration value="402000" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="sts" minOccurs="0">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="stsL" minOccurs="0" maxOccurs
="unbounded">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="s" type="xsd:decimal" minOccurs="0"
/>
                            <xsd:element name="st" type="xsd:string" minOccurs="0"
/>
                                </xsd:sequence>
                            </xsd:complexType>
                        </xsd:element>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="msrs" minOccurs="0">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="mL" minOccurs="0" maxOccurs="unbounded">
                            <xsd:complexType>
                                <xsd:sequence>
                                    <xsd:element name="s" type="xsd:decimal" minOccurs="0"
/>
                                        <xsd:element name="dt" type="xsd:dateTime"
minOccurs="0" />
                                            <xsd:element name="q" type="xsd:decimal" minOccurs="0"
/>
                                                <xsd:element name="ue" minOccurs="0">
                                                    <xsd:simpleType>
                                                        <xsd:restriction base="xsd:string">
                                                            <xsd:enumeration value="userEdited" />
                                                        </xsd:restriction>
                                                    </xsd:simpleType>
                                                </xsd:element>
                                                <xsd:element name="fc" minOccurs="0">
                                                    <xsd:simpleType>
                                                        <xsd:restriction base="xsd:string">
                                                            <xsd:enumeration value="301000" />
                                                            <xsd:enumeration value="901000" />
                                                            <xsd:enumeration value="501000" />
                                                            <xsd:enumeration value="101000" />
                                                            <xsd:enumeration value="100000" />
                                                            <xsd:enumeration value="201000" />
                                                            <xsd:enumeration value="401000" />
                                                            <xsd:enumeration value="402000" />
                                                        </xsd:restriction>
                                                    </xsd:simpleType>
                                                </xsd:element>
                                            <xsd:element name="sts" minOccurs="0">
                                                <xsd:complexType>
                                                    <xsd:sequence>
                                                        <xsd:element name="stsL" minOccurs="0"
maxOccurs="unbounded">
                                                            <xsd:complexType>
                                                                <xsd:sequence>
                                                                    <xsd:element name="s" type="xsd:decimal"
minOccurs="0" />
                                                                        <xsd:element name="st" type="xsd:string"
minOccurs="0" />
                                                                            </xsd:sequence>
                                                                        </xsd:complexType>
                                                                    </xsd:element>
                                                                </xsd:sequence>
                                                            </xsd:complexType>
                                                        </xsd:element>
                                                    </xsd:sequence>
                                                </xsd:complexType>
                                            </xsd:element>
                                        </xsd:sequence>
                                    </xsd:complexType>
                                </xsd:sequence>
                            </xsd:complexType>
                        </xsd:element>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

```

```
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="rawData" type="xsd:anyType" minOccurs="0"
maxOccurs="unbounded" />
  <xsd:element name="processData" minOccurs="0">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="isShiftedStartEnd" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="no" />
              <xsd:enumeration value="yes" />
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="isShiftedIntervals" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="no" />
              <xsd:enumeration value="yes" />
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="isErrorEncountered" minOccurs="0">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="no" />
              <xsd:enumeration value="yes" />
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="servicePointId" type="xsd:string"
minOccurs="0" />
        <xsd:element name="installationConstant" type="xsd:decimal"
minOccurs="0" />
        <xsd:element name="deviceId" type="xsd:string" minOccurs="0" />
        <xsd:element name="logs" minOccurs="0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="logsList" minOccurs="0"
maxOccurs="unbounded">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="logsEntry" minOccurs="0">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="sequence" type="xsd:decimal"
minOccurs="0" />
                          <xsd:element name="mo" type="xsd:string"
minOccurs="0" />
                          <xsd:element name="pkValue1" type="xsd:string"
minOccurs="0" />
                          <xsd:element name="pkValue2" type="xsd:string"
minOccurs="0" />
                          <xsd:element name="pkValue3" type="xsd:string"
minOccurs="0" />
                          <xsd:element name="pkValue4" type="xsd:string"
minOccurs="0" />
```

```

minOccurs="0" />
<xsd:element name="pkValue5" type="xsd:string"
<xsd:element name="logEntryType" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="toDos" />
      <xsd:enumeration value="created" />
      <xsd:enumeration
value="statusTransitionError" />
      <xsd:enumeration value="exception" />
      <xsd:enumeration value="statusTransition" /
>
      <xsd:enumeration value="system" />
      <xsd:enumeration value="todo" />
      <xsd:enumeration value="userDetails" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="logDateTime"
type="xsd:dateTime" minOccurs="0" />
<xsd:element name="boStatus" type="xsd:string"
minOccurs="0" />
<xsd:element name="description"
type="xsd:string" minOccurs="0" />
<xsd:element name="user" type="xsd:string"
minOccurs="0" />
<xsd:element name="logMessage" type="xsd:string"
minOccurs="0" />
<xsd:element name="characteristicType"
type="xsd:string" minOccurs="0" />
<xsd:element name="characteristicValue"
type="xsd:string" minOccurs="0" />
<xsd:element name="adhocValue" type="xsd:string"
minOccurs="0" />
<xsd:element name="fkValue1" type="xsd:string"
minOccurs="0" />
<xsd:element name="fkValue2" type="xsd:string"
minOccurs="0" />
<xsd:element name="fkValue3" type="xsd:string"
minOccurs="0" />
<xsd:element name="fkValue4" type="xsd:string"
minOccurs="0" />
<xsd:element name="fkValue5" type="xsd:string"
minOccurs="0" />
<xsd:element name="messageCategory"
type="xsd:decimal" minOccurs="0" />
<xsd:element name="messageNumber"
type="xsd:decimal" minOccurs="0" />
<xsd:element name="messageParm1"
type="xsd:string" minOccurs="0" />
<xsd:element name="messageParm2"
type="xsd:string" minOccurs="0" />
<xsd:element name="messageParm3"
type="xsd:string" minOccurs="0" />
<xsd:element name="messageParm4"
type="xsd:string" minOccurs="0" />
<xsd:element name="messageParm5"
type="xsd:string" minOccurs="0" />
<xsd:element name="messageParm6"
type="xsd:string" minOccurs="0" />
<xsd:element name="messageParm7"
type="xsd:string" minOccurs="0" />
<xsd:element name="messageParm8"
type="xsd:string" minOccurs="0" />
<xsd:element name="messageParm9"
type="xsd:string" minOccurs="0" />
</xsd:sequence>
</xsd:complexType>

```

```

        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="deviceEventTypes" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="deviceEventTypesList" minOccurs="0"
maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="deviceEventType" type="xsd:string"
minOccurs="0" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="boStatus" type="xsd:string" minOccurs="0" />
<xsd:element name="statusReason" type="xsd:string" minOccurs="0" />
<xsd:element name="bo" type="xsd:string" />
<xsd:element name="creationDateTime" type="xsd:dateTime" minOccurs="0"
/>
<xsd:element name="boStatusDateTime" type="xsd:dateTime" minOccurs="0"
/>
<xsd:element name="isTraceOn" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="no" />
      <xsd:enumeration value="yes" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="isIntervalDateTimePopulated">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="no" />
      <xsd:enumeration value="yes" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="isReprocessPerformed" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="no" />
      <xsd:enumeration value="yes" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="serviceProvider" type="xsd:string" minOccurs="0" />
<xsd:element name="serviceProviderExternalId" type="xsd:string"
minOccurs="0" />
<xsd:element name="fromDateTime" type="xsd:dateTime" minOccurs="0" />
<xsd:element name="toDateTime" type="xsd:dateTime" minOccurs="0" />
<xsd:element name="timeZone" type="xsd:string" minOccurs="0" />
<xsd:element name="isAutomatedRetry" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="no" />
      <xsd:enumeration value="yes" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

```

```
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="retryUntilDateTime" type="xsd:dateTime"
minOccurs="0" />
      <xsd:element name="version" type="xsd:decimal" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="dateTimeTagFormat" type="xsd:string" fixed="xsd"
use="required" />
    <xsd:attribute name="transactionType">
      <xsd:simpleType>
        <xsd:restriction base="xsd:token">
          <xsd:enumeration value="RWOV" />
          <xsd:enumeration value="FADD" />
          <xsd:enumeration value="FUPD" />
          <xsd:enumeration value="DEL" />
          <xsd:enumeration value="UPD" />
          <xsd:enumeration value="ADD" />
          <xsd:enumeration value="READ" />
          <xsd:enumeration value="REPL" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
```

Device Event XML Format

This section describes the XML format used for inbound device events. This includes an example, descriptions of the individual XML elements, and the device event XML schema based on the D1-DeviceEventSeeder business object.

Example - Device Event

```
<D1-DeviceEventSeeder>
  <externalSenderId>L+G</externalSenderId>
  <deviceIdentifierNumber>GD_LL_SN100</deviceIdentifierNumber>
  <externalEventName>GD_LL_TAMPER_INDICATION</externalEventName>

  <rawEventInformation>D~Meter~GD_LL_SN100~GD_LL_ELECTRIC~1342395718~3~GD_HeadEnd_Power_Off~2010-09-09T13:11:41.0000000-05:00~Alert~Tamper indication on serial number GD_LL_SN100.~2010-09-09T13:11:41.0000000-05:00</rawEventInformation>
  <eventDateTime>2010-09-09-13.11.41</eventDateTime>
  <externalSourceIdentifier>EVENT_test.lg</externalSourceIdentifier>
  <eventInformation>
    <externalEventCategory>3</externalEventCategory>
    <externalEventSeverity>Alert</externalEventSeverity>
    <externalDeviceType>Meter</externalDeviceType>
    <externalServiceLocationId>GD_LL_ELECTRIC</externalServiceLocationId>
    <externalCommunicationModuleIdentifier>1342395718</externalCommunicationModuleIdentifier>
    </externalGatewayIdentifier>
    <externalStatusValue>Tamper indication on serial number GD_LL_SN100.</externalStatusValue>
    <externalStatusDateTime>2010-09-09-13.11.41</externalStatusDateTime>
    </sourceTimeZone>
    </timeZone>
    </dateTimesInStandard>
  </eventInformation>
</D1-DeviceEventSeeder>
```

Element Descriptions - Device Events

The table below provides descriptions of the elements used in the device event XML format.

Element	Description
<{SERVICE_NAME}>	Root element containing a device event. This element should match the name of the inbound service used to import device events.
<externalSenderId>	Id of the external system. Used to identify the head-end system from which the event is being sent.
<deviceIdentifierNumber>	The identifier number of the device that experienced the event.
<externalEventName>	The name of the event as defined by the external system. This will be mapped to a standard name via a device mapping business object.
<rawEventInformation>	A string containing information about the event from the external system.
<eventDateTime>	The date and time of the event.

Element	Description
<externalSourceIdentifier>	The name of the file containing the device event.
<eventInformation>	Element containing specific information about the event
<externalEventCategory>	The event category as defined by the external system.
<externalEventSeverity>	The event severity as defined by the external system.
<externalDeviceType>	The device type as defined by the external system. Can be one of the following: <ul style="list-style-type: none"> • Meter • Collector • Router
<externalServiceLocationId>	The service location for the event defined by the external system.
<externalCommunicationModuleIdentifier>	The identifier for the communication module associated with the device.
</externalGatewayIdentifier>	The identifier for the gateway associated with the device
<externalStatusValue>	Optional information related to the event.
<externalStatusDateTime>	Date and time at which optional information (specified in the <externalStatusValue> element was recorded.
</sourceTimeZone>	The source time zone for the event.
</timeZone>	The target time zone for the event.
</dateTimesInStandard>	A flag that indicates if date and times have been adjusted to Standard time

Schema - Device Event Seeder (D1-DeviceEventSeeder) Business Object

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ouaf="http://
ouaf.oracle.com/" targetNamespace="http://oracle.com/D1-DeviceEventSeeder.xsd"
elementFormDefault="qualified">
  <xsd:import namespace="http://ouaf.oracle.com/" />
  <xsd:element name="D1-DeviceEventSeeder">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="deviceEventId" type="xsd:string" minOccurs="0" />
        <xsd:element name="bo" type="xsd:string" minOccurs="0" />
        <xsd:element name="boStatus" type="xsd:string" minOccurs="0" />
        <xsd:element name="sender" type="xsd:string" minOccurs="0" />
        <xsd:element name="externalSenderId" type="xsd:string" minOccurs="0" />
        <xsd:element name="deviceEventType" type="xsd:string" minOccurs="0" />
        <xsd:element name="externalEventName" type="xsd:string" minOccurs="0" /
    >
  </xsd:element>
</xsd:schema>

```

```

        <xsd:element name="eventDateTime" type="xsd:dateTime" />
        <xsd:element name="eventEndDateTime" type="xsd:dateTime" minOccurs="0"
/>
        <xsd:element name="deviceId" type="xsd:string" minOccurs="0" />
        <xsd:element name="creationDateTime" type="xsd:dateTime" minOccurs="0"
/>
        <xsd:element name="statusUpdateDateTime" type="xsd:dateTime"
minOccurs="0" />
        <xsd:element name="statusReason" type="xsd:string" minOccurs="0" />
        <xsd:element name="rawEventInformation" type="xsd:anyType"
minOccurs="0" maxOccurs="unbounded" />
        <xsd:element name="externalSourceIdentifier" type="xsd:string"
minOccurs="0" />
        <xsd:element name="eventInformation" minOccurs="0">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="externalEventIdentifier" type="xsd:string"
minOccurs="0" />
                    <xsd:element name="externalEventCategory" type="xsd:string"
minOccurs="0" />
                    <xsd:element name="externalEventSeverity" type="xsd:string"
minOccurs="0" />
                    <xsd:element name="externalDeviceType" type="xsd:string"
minOccurs="0" />
                    <xsd:element name="externalServiceLocationId" type="xsd:string"
minOccurs="0" />
                    <xsd:element name="externalCommunicationModuleIdentifier"
type="xsd:string" minOccurs="0" />
                    <xsd:element name="externalGatewayIdentifier" type="xsd:string"
minOccurs="0" />
                    <xsd:element name="externalStatusValue" type="xsd:string"
minOccurs="0" />
                    <xsd:element name="externalStatusDateTime" type="xsd:dateTime"
minOccurs="0" />
                    <xsd:element name="sourceTimeZone" type="xsd:string"
minOccurs="0" />
                    <xsd:element name="timeZone" type="xsd:string" minOccurs="0" />
                    <xsd:element name="dateTimesInStandard" minOccurs="0" />
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:enumeration value="no" />
                            <xsd:enumeration value="yes" />
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="version" type="xsd:decimal" minOccurs="0" />
        <xsd:element name="deviceIdIdentifierNumber" type="xsd:string"
minOccurs="0" />
        <xsd:element name="newDeviceEvent" type="xsd:string" minOccurs="0" />
        <xsd:element name="processData" minOccurs="0">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="errorEncountered" minOccurs="0">
                        <xsd:simpleType>
                            <xsd:restriction base="xsd:string">
                                <xsd:enumeration value="no" />
                                <xsd:enumeration value="yes" />
                            </xsd:restriction>
                        </xsd:simpleType>
                    </xsd:element>
                    <xsd:element name="dateTimesInStandard" minOccurs="0">
                        <xsd:simpleType>
                            <xsd:restriction base="xsd:string">
                                <xsd:enumeration value="no" />
                                <xsd:enumeration value="yes" />
                            </xsd:restriction>
                        </xsd:simpleType>
                    </xsd:element>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

```

```

        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
    <xsd:element name="logs" minOccurs="0">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="logsList" minOccurs="0"
maxOccurs="unbounded">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="logsEntry" minOccurs="0">
                                <xsd:complexType>
                                    <xsd:sequence>
                                        <xsd:element name="sequence" type="xsd:decimal"
minOccurs="0" />
                                        <xsd:element name="mo" type="xsd:string"
minOccurs="0" />
                                        <xsd:element name="pkValue1" type="xsd:string"
minOccurs="0" />
                                        <xsd:element name="pkValue2" type="xsd:string"
minOccurs="0" />
                                        <xsd:element name="pkValue3" type="xsd:string"
minOccurs="0" />
                                        <xsd:element name="pkValue4" type="xsd:string"
minOccurs="0" />
                                        <xsd:element name="pkValue5" type="xsd:string"
minOccurs="0" />
                                        <xsd:element name="logEntryType" minOccurs="0">
                                            <xsd:simpleType>
                                                <xsd:restriction base="xsd:string">
                                                    <xsd:enumeration value="toDos" />
                                                    <xsd:enumeration value="created" />
                                                    <xsd:enumeration
value="statusTransitionError" />
                                                    <xsd:enumeration value="exception" />
                                                    <xsd:enumeration value="statusTransition" />
                                                    <xsd:enumeration value="system" />
                                                    <xsd:enumeration value="todo" />
                                                    <xsd:enumeration value="userDetails" />
                                                </xsd:restriction>
                                            </xsd:simpleType>
                                        </xsd:element>
                                        <xsd:element name="logDateTime"
type="xsd:dateTime" minOccurs="0" />
                                        <xsd:element name="boStatus" type="xsd:string"
minOccurs="0" />
                                        <xsd:element name="description"
type="xsd:string" minOccurs="0" />
                                        <xsd:element name="user" type="xsd:string"
minOccurs="0" />
                                        <xsd:element name="logMessage" type="xsd:string"
minOccurs="0" />
                                        <xsd:element name="characteristicType"
type="xsd:string" minOccurs="0" />
                                        <xsd:element name="characteristicValue"
type="xsd:string" minOccurs="0" />
                                        <xsd:element name="adhocValue" type="xsd:string"
minOccurs="0" />
                                        <xsd:element name="fkValue1" type="xsd:string"
minOccurs="0" />
                                        <xsd:element name="fkValue2" type="xsd:string"
minOccurs="0" />
                                        <xsd:element name="fkValue3" type="xsd:string"
minOccurs="0" />
                                        <xsd:element name="fkValue4" type="xsd:string"
minOccurs="0" />

```

```
minOccurs="0" />
                                <xsd:element name="fkValue5" type="xsd:string"
                                <xsd:element name="messageCategory"
type="xsd:decimal" minOccurs="0" />
                                <xsd:element name="messageNumber"
type="xsd:decimal" minOccurs="0" />
                                <xsd:element name="messageParm1"
type="xsd:string" minOccurs="0" />
                                <xsd:element name="messageParm2"
type="xsd:string" minOccurs="0" />
                                <xsd:element name="messageParm3"
type="xsd:string" minOccurs="0" />
                                <xsd:element name="messageParm4"
type="xsd:string" minOccurs="0" />
                                <xsd:element name="messageParm5"
type="xsd:string" minOccurs="0" />
                                <xsd:element name="messageParm6"
type="xsd:string" minOccurs="0" />
                                <xsd:element name="messageParm7"
type="xsd:string" minOccurs="0" />
                                <xsd:element name="messageParm8"
type="xsd:string" minOccurs="0" />
                                <xsd:element name="messageParm9"
type="xsd:string" minOccurs="0" />
                                </xsd:sequence>
                                </xsd:complexType>
                                </xsd:element>
                                </xsd:sequence>
                                </xsd:complexType>
                                </xsd:element>
                                </xsd:sequence>
                                </xsd:complexType>
                                </xsd:element>
                                </xsd:sequence>
                                </xsd:complexType>
                                </xsd:element>
                                </xsd:sequence>
                                <xsd:attribute name="dateTimeTagFormat" type="xsd:string" fixed="xsd"
use="required" />
                                <xsd:attribute name="transactionType">
                                <xsd:simpleType>
                                <xsd:restriction base="xsd:token">
                                <xsd:enumeration value="RWOV" />
                                <xsd:enumeration value="FADD" />
                                <xsd:enumeration value="FUPD" />
                                <xsd:enumeration value="DEL" />
                                <xsd:enumeration value="UPD" />
                                <xsd:enumeration value="ADD" />
                                <xsd:enumeration value="READ" />
                                <xsd:enumeration value="REPL" />
                                </xsd:restriction>
                                </xsd:simpleType>
                                </xsd:attribute>
                                </xsd:complexType>
                                </xsd:element>
</xsd:schema>
```

Usage and Event Import - Message Driven Bean Configuration

This section describes the steps for configuring the Message Driven Bean (MDB) feature of Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway to listen to inbound JMS messages. This feature is used when importing usage reading and device events from head-end systems. This section includes:

- **JMS Configuration**
- **Message Driven Bean Configuration**
- **Notification Queue Configuration**

JMS Configuration

JMS configuration involves setting up JMS queues which will receive inbound usage readings and device events. The JMS queues need to be created first on the application server where the OSB component is deployed. This server is referred to as remote server in the sections below. In the following section the JMS queue on the remote server is assumed to be created with the name **DestinationQueueWatch-CM**.

Note: The JMS changes described in the following sections are not persistent during patches or upgrades. They will need to be re-created after applying any patches or upgrades to Oracle Utilities Smart Grid Gateway. It is recommended to keep a backup of the \$SPLBASE/splapp/config.xml file.

Create a new JMS module

Log in to the Oracle Utilities Smart Grid Gateway Weblogic console and create a JMS Module with an appropriate name. Specify the following values for this JMS module:

- **Name:** the name of JMS module. For example, JMSModule-CM
- **Target:** the name of the target server where the Oracle Utilities Smart Grid Gateway application is running. This should be specified as myserver.

Create a Foreign JMS server

Create a Foreign JMS server under the JMS module created in the above step. Specify the following values for this foreign JMS server:

- **Name:** Name of the foreign server. For example, JMSFAServer-CM
- **Target:** This should be specified as myserver
- **JNDI Initial Context Factory:** This should be specified as weblogic.jndi.WLInitialContextFactory
- **JNDI Connection URL:** The URL of the server where OSB is deployed. For example: t3://osbserver:7001
- **JNDI Properties Credential:** Password for the OSB server user.
- **JNDI Properties:** The java.naming.security.principal additional property should be specified and set to the OSB server user. For example, java.naming.security.principal=weblogic

Create a Foreign Destination

Create a Foreign destination for each remote queue. Specify the following values for this foreign destination:

- **Name:** Name of foreign destination. For instance, DestinationQueue-CM
- **Local JNDI Name:** Local JNDI name for the foreign JMS queue. For example, ForeignDestinationQueue-CM

- **Remote JNDI Name:** JNDI name of the queue on the remote server. For example, DestinationQueueWatch-CM

Create a Remote Connection Factory

Create a remote connection factory for the foreign JMS server. Specify the following values for this remote connection factory:

- **Name:** Name of remote connection factory. For example, DestinationQueueConnectionFactory-CM
- **Local JNDI Name:** Local JNDI name for the Remote Connection Factory. For example, ForeignDestinationQueueConnectionFactory-CM
- **Remote JNDI Name:** JNDI name of the JMS Connection Factory on the remote server. For example, weblogic.jms.XAConnectionFactory

Message Driven Bean Configuration

Configuration of message driven beans (MDB) involved modifying the **ejb-jar.xml** and **weblogic-ejb-jar.xml** configuration files delivered with Oracle Utilities Smart Grid Gateway. It is recommended that instead of modifying these files directly you create “Customer Modification” (CM) versions of these files to make changes to these configuration files. This ensures that your modifications are not overwritten by future application patches.

The following section describes the changes required in the CM files for configuring the MDBs to read from the foreign JMS queues set up in the steps above (see **JMS Configuration** on page 15-53). This requires creating the following files under \$SPLEBASE/templates -

- cm_ejb-jar.xml.wls.jms_1.include
- cm_ejb-jar.xml.wls.jms_2.include
- cm_weblogic-ejb-jar.xml.jms.include.

Note: After making these changes the initialSetup script needs to be run and Oracle Utilities Smart Grid Gateway application needs to be redeployed. However the initialSetup script will overwrite the JMS configuration changes made in the steps above. So it is recommended to keep a backup of the \$SPLEBASE/splapp/config.xml file before running this script.

Changes to cm_ejb-jar.xml.wls.jms_1.include

Below is an example of the cm_ejb-jar.xml.wls.jms_1.include file:

```
<message-driven>
  <description>MDB for DestinationQueue-CM</description>
  <display-name>DestinationQueueWatcher-CM</display-name>
  <ejb-name>DestinationQueueWatch-CM</ejb-name>
  <ejb-class>com.splwg.ejb.mdb.MessageProcessor</ejb-class>
  <messaging-type>javax.jms.MessageListener</messaging-type>
  <transaction-type>Bean</transaction-type>
  <message-destination-type>javax.jms.Queue</message-destination-type>
</message-driven>
```

The values specified in the above file include the following:

- **ejb-name:** This is the name of the MDB.

Changes to cm_ejb-jar.xml.wls.jms_2.include

Below is an example of the cm_ejb-jar.xml.wls.jms_2.include file:

```
<assembly-descriptor>
  <security-role>
    <role-name>cisusers</role-name>
  </security-role>
  <container-transaction>
```

```

<method>
  <ejb-name>DestinationQueueWatch-CM</ejb-name>
  <method-name>onMessage</method-name>
</method>
<trans-attribute>NotSupported</trans-attribute>
</container-transaction>
</assembly-descriptor>

```

The values specified in the above file include the following:

- **ejb-name:** This is the name of the MDB.

Changes to cm_weblogic-ejb-jar.xml.jms.include

Below is an example of the cm_weblogic-ejb-jar.xml.jms.include file:

```

<weblogic-enterprise-bean>
  <ejb-name>DestinationQueueWatch-CM</ejb-name>
  <message-driven-descriptor>
    <pool>
      <max-beans-in-free-pool>5</max-beans-in-free-pool>
      <initial-beans-in-free-pool>1</initial-beans-in-free-pool>
    </pool>
    <destination-jndi-name>ForeignDestinationQueue-CM</destination-jndi-name>
    <connection-factory-jndi-name>ForeignConnectionFactory-CM</connection-
factory-jndi-name>
  </message-driven-descriptor>
</weblogic-enterprise-bean>

```

The values specified in the above file include the following:

- **ejb-name:** This should be the name of the MDB as specified in ejb-jar.xml.
- **destination-jndi-name:** This should be the JNDI name of the foreign destination as provided in JMS module ' Foreign server ' Foreign destination ' Local JNDI name.
- **connection-factory-jndi-name:** This should be the JNDI name of the connection factory as provided in JMS module ' Foreign server ' Remote Connection Factory ' Local JNDI name.

Notification Queue Configuration

Payload statistics and payload summary records must be submitted sequentially in order for them to be processed correctly. To prevent them from being processed at the same time, you should set the number of notification queue polling threads to 1. Follow these steps to configure the number of notification queue threads:

1. Log in to the WebLogic Server Administration Console.
2. Under Helpful Tools, click **Configure Applications**.
3. Click on **SPLService**.
4. Click on the NotificationQueue link. for the EJB that you want to configure.
5. Go to the Configuration tab.
6. In the Change Center, click **Lock & Edit**.
7. Specify the new value of polling threads in **Max Beans in Free Pool**.
8. Click **Save**.
9. Click **Release Configuration**.
10. Restart the OUAF WebLogic instance.

Chapter 16

Configuration Examples and Sample Implementation

This chapter provides configuration examples and a simple example implementation of Oracle Utilities Meter Data Management.

- **Configuration Examples**
- **Sample Implementation**

Configuration Examples

This section provides examples of how to create and configure data to meet scenarios and requirements that might be typically encountered when configuring the system as part of an implementation.

For each scenario described, these examples provide a table that lists the specific objects (or types of data) that need to be created (and the specific business object used), and the values for the fields/parameters used by each type of data.

Notes:

- These examples do not contain sample values for every field for every object. Only the data that plays a significant part in meeting the scenario requirements is included. Refer to the **Base Package Objects** sections in the *Oracle Utilities Service and Measurement Data Foundation User's Guide* and the *Oracle Utilities Meter Data Management User's Guide* for more information about the fields/parameters used by each type of object.
- These examples all use business objects provided with the base package.

General Data

This section provides examples of data configuration for general data types that used by other entities in the system.

Units of Measure Example

Scenario: Creates units of measure to represent related UOMs, such as KWH and MWH.

Object	Data and Field Values
Unit of Measure: KWH Business Object: D1-UnitOfMeasure	<ul style="list-style-type: none"> • Unit of Measure: KWH • Description: Kilowatt Hour • Shorthand Description: KWH • Service Type: Electric • Allowed on Measuring Component: Yes • Measures Peak Quantity: No • Magnitude: 1 • Base Unit of Measure: N/A
Unit of Measure: MWH Business Object: D1-UnitOfMeasure	<ul style="list-style-type: none"> • Unit of Measure: MWH • Description: Megawatt Hour • Shorthand Description: MWH • Service Type: Electric • Allowed on Measuring Component: Yes • Measures Peak Quantity: No • Magnitude: 100 • Base Unit of Measure: Kilowatt Hour

Time Of Use Examples

Scenario: Create the data needed to map usage data for two time of use periods as follows:

- **On Peak:** Monday - Friday - 9:00 AM to 5:00 PM
- **Off Peak:** All other times

Object	Data and Field Values
Time of Use: On Peak Business Object: D1-TimeOfUse	<ul style="list-style-type: none"> • Time of Use: ON • Description: On Peak • Color: #CCFFCC • Priority: 1
Time of Use: Off Peak Business Object: D1-TimeOfUse	<ul style="list-style-type: none"> • Time of Use: OFF • Description: Off Peak • Color: #008000 • Priority: 2
TOU Group Business Object: D1-TOUGroup	<ul style="list-style-type: none"> • TOU Group: ON-OFF • Description: On / Off • TOU Section (Priority / TOU): <ul style="list-style-type: none"> • 1 / On Peak • 2 / Off Peak
TOU Map Template Business Object: D2-TOUMapTemplate	<ul style="list-style-type: none"> • TOU Map Template: WORKHOURS • Description: Weekday - Office Hours • TOU Group: On / Off • Default TOU: Off Peak • Work Calendar: US Work Days • Holiday TOU: Off Peak • Seconds Per Interval: 00:30:00 • TOU Schedule Section: <ul style="list-style-type: none"> • Start Date: 01-01 • End Date: 12-31 • Start Weekday: Monday • End Weekday: Friday • Start Time: 09:00:00 AM • Stop Time: 05:00:00 PM • TOU: On Peak

Object	Data and Field Values
TOU Map Type Business Object: D2-TOUMapType	<ul style="list-style-type: none"> • TOU Map Type: WORKHOURS • Description: Weekday - Office Hours • TOU Map Business Object: Time of Use Map • Time Zone: US Pacific Time • Seconds Per Interval: 00:30:00 • Default TOU Map Template: Weekday - Office Hours • Override TOU Map Templates: N/A
TOU Map Business Object: D2-TOUMap	<ul style="list-style-type: none"> • Description: Weekday - Office Hours • TOU Map Type: Weekday - Office Hours • Status: Active • Override TOU Map Templates: N/A • Dynamic Option / Dynamic TOU Map Section: N/A

Dynamic Options

Scenario: Create the data needed to represent a critical peak pricing period for April 10, 2013 from 8:00 AM to 9:30 AM.

Object	Data and Field Values
Dynamic Option Type Business Object: D2-DynamicOptionType	<ul style="list-style-type: none"> • Dynamic Option Type: AM_CPP • Description: AM Critical Peak Period • Dynamic Option Business Object: Dynamic Option • Time Zone: US Pacific Time
Dynamic Option Business Object: D2-DynamicOption	<ul style="list-style-type: none"> • Description: AM Critical Peak Period • Dynamic Option Type: AM Critical Peak Period • Status: Active
Dynamic Option Event Business Object: D2-DynamicOptionEvent	<ul style="list-style-type: none"> • Dynamic Option ID: AM Critical Peak Period / Active • Status: Frozen • Start Date/Time: 04-10-2013 08:00:00AM • End Date/Time: 04-10-2013 09:30:00AM

Service Quantity Identifier Example

Scenario: Create service quantity identifiers to represent service quantities for the following specific types of a particular unit measure:

- Consumed KWH
- Generated KWH

Object	Data and Field Values
Service Quantity Identifier: Consumed KWH Business Object: D1-ServiceQuantityIdentifier	<ul style="list-style-type: none"> • Service Quantity Identifier: CONSUMED_KWH • Description: Consumed KWH • Decimal Positions: 2
Service Quantity Identifier: Generated KWH Business Object: D1-ServiceQuantityIdentifier	<ul style="list-style-type: none"> • Service Quantity Identifier: GENERATED_KWH • Description: Generated KWH • Decimal Positions: 2

Service Provider Examples

This section provides configuration examples related to service providers, including head-end systems, external applications, and processing methods used by each.

Head-End System

Scenario: Create a service provider to represent a "Meters-R-Us" head-end system as follows:

- Uses serial numbers for AMI device IDs
- Uses channel IDs to represent registers and channels on their meters

Object	Data and Field Values
External System	<ul style="list-style-type: none"> • External System: METERSUS • Description: Meters-R-Us
Service Provider: Head-End System Business Object: D1-HeadEndSystem	<ul style="list-style-type: none"> • Service Provider: METERSUS • Description: Meters-R-Us • External System: Meters-R-Us • AMI Device ID Type: Serial Number • AMI Measuring Component ID Type: Channel ID

External Application

Scenario: Create a service provider to represent Oracle Utilities Customer Care and Billing, as follows

- Uses an external ID for service point IDs
- Uses badge numbers for device IDs
- Uses channel IDs to represent registers and channels on their meters

Object	Data and Field Values
External System	<ul style="list-style-type: none"> • External System: CCB • Description: Customer Care and Billing
Service Provider: External Application Business Object: D1-ExternalApplication	<ul style="list-style-type: none"> • Service Provider: CCB • Description: Customer Care and Billing • External System: Customer Care and Billing • Utility Device ID Type: Badge Number • Utility Measuring Component ID Type: Channel ID • Utility Service Point ID Type: External ID

Processing Methods

Scenario: Configure processing methods for the "Meters-R-Us" head-end system (described above) to receive usage from the head-end system as follows:

- The default type of usage received is interval.
- Scalar usage is received for a single specific type of meter (Daily Scalar),

Object	Data and Field Values
Processing Method: Initial Measurement Creation Business Object: D1-HowToCreateMCInformation	<ul style="list-style-type: none"> • Service Provider: Meters-R-Us • Processing Role: Initial Measurement Creation • Description: Initial Measurement Creation • Status: Active • Default Processing Method (Business Object): D1-InitialLoadIMDInterval • Override Processing Method: <ul style="list-style-type: none"> • Measuring Component Type: Daily Scalar • Business Object: D1-InitialLoadIMDScalar

Scenario: Configure processing methods for the "Meters-R-Us" head-end system (described above) to receive device events from the head-end system as follows:

- The Meters R Us head-end system's name for a tampering event is "Tamper Alarm".
- The "standard" event name for tampering event is "DeviceTamperAlert / Tamper Alert".

- Device event mapping for this service provider uses the base package extendable lookup.

Object	Data and Field Values
Standard Event Name Extendable Lookup Business Object: D1- StdEventNameLookup	<ul style="list-style-type: none"> • Standard Event Name: DeviceTamperAlert • Description: Tamper Alert • Usage Flag: Active
Device Event Mapping Extendable Lookup Business Object: D1- DeviceEventMappingLookup	<ul style="list-style-type: none"> • Head-End System Event Name: Tamper Alarm • Description: Tamper Alarm • Standard Event Name: Tamper Alert
Processing Method: Initial Measurement Creation Business Object: D1- HowToCreateMCInformation	<ul style="list-style-type: none"> • Service Provider: Meters-R-Us • Processing Role: Device Event Mapping • Description: Meters R Us Device Event Mapping • Status: Active • Default Processing Method (Business Object): D1-DeviceEventMappingLookup

Scenario: Configure processing methods for the "Customer Care and Billing" external application (described above) to receive usage requests:

Object	Data and Field Values
Processing Method: Usage Transaction Creation Business Object: D2- HowToCreateUSInformation	<ul style="list-style-type: none"> • Service Provider: Customer Care and Billing • Processing Role: Usage Transaction Creation • Description: Usage Transaction Creation • Status: Active • Default Processing Method (Business Object): D2-UsageTransaction • Override Processing Method: N/A

Devices

This section provides configuration examples for setting up devices, including meters, items, and communication components.

Smart Meter Examples

Scenario: Create an interval smart meter such that usage and/or events for the meter can be received into the system, as follows:

- Interval Data Size: 60 Minute
- Head-End System: Meters-R-Us (see above)

Object	Data and Field Values
Measuring Component Type Business Object: D1-IntervalChannelTypePhysical	<ul style="list-style-type: none"> • Measuring Component Type: KWH-60MIN • Description: KWH - 60 Minutes • Measuring Component Business Object: Interval Channel • Measurement Business Object: Measurement • Service Type: Electric • Consumptive / Subtractive: Consumptive • Seconds Per Interval: 01:00:00
Device Configuration Type Business Object: D1-DeviceConfigurationType	<ul style="list-style-type: none"> • Device Configuration Type: KWH-INTERVAL • Description: KWH - Interval Channel • Device Configuration Business Object: Device Configuration • Service Type: Electric • Valid Measuring Component Types: <ul style="list-style-type: none"> • KWH - 60 Minutes
Device Type Business Object: D1-SmartMeterType	<ul style="list-style-type: none"> • Device Type: ELEC-SMART-INT-KWH • Description: Electric Smart Meter - Interval KWH • Device Business Object: Smart Meter • Service Type: Electric • Head-End System (Fallback): Meters-R-Us • Device Classification: Meter • Valid Device Configuration Types: <ul style="list-style-type: none"> • KWH - Interval Channel • Valid Head-End Systems: <ul style="list-style-type: none"> • Meters-R-Us

Object	Data and Field Values
Device Business Object: D1-SmartMeter	<ul style="list-style-type: none"> • Device Type: Electric Smart Meter - Interval KWH • Serial Number: 12345 • Head-End System: Meters-R-Us • Status: Active
Device Configuration Business Object: D1-DeviceConfiguration	<ul style="list-style-type: none"> • Device Configuration Type: KWH - Interval Channel • Device: 12345 / Electric Smart Meter - Interval KWH / Meters-R-Us / Active • Effective Date/Time: 01-01-2014 12:00:00AM • Time Zone: US Pacific Time • External ID: 54321
Measuring Component Business Object: D1-IntervalChannel	<ul style="list-style-type: none"> • Measuring Component Type: KWH - 60 Minutes • Device Configuration: Electric Smart Meter - Interval KWH / Effective Date/Time: 01-01-2014 12:00:00AM / KWH - Interval Channel / 1 Measuring Component(s) / Active • How To Use: Additive • Channel ID: 1

Scenario: Create a scalar auto-read register smart meter such that usage and/or events for the meter can be received into the system, as follows:

- Head-End System: Meters-R-Us (see above)

Object	Data and Field Values
Measuring Component Type Business Object: D1-AutoReadRegisterType	<ul style="list-style-type: none"> • Measuring Component Type: KWH-AUTO-READ • Description: KWH - Auto Read • Measuring Component Business Object: Register - Auto Read • Measurement Business Object: Measurement • Service Type: Electric • Consumptive / Subtractive: Subtractive • Read Method: Automatic

Object	Data and Field Values
Device Configuration Type Business Object: D1-DeviceConfigurationType	<ul style="list-style-type: none"> • Device Configuration Type: KWH-SCALAR • Description: KWH - Scalar • Device Configuration Business Object: Device Configuration • Service Type: Electric • Valid Measuring Component Types: <ul style="list-style-type: none"> • Auto Read - Electric - KWH
Device Type Business Object: D1-SmartMeterType	<ul style="list-style-type: none"> • Device Type: ELEC_SMART_SCL_KWH • Description: Electric Smart Meter - Scalar KWH • Device Business Object: Smart Meter • Service Type: Electric • Head-End System (Fallback): Meters-R-Us • Device Classification: Meter • Valid Device Configuration Types: <ul style="list-style-type: none"> • KWH - Scalar • Valid Head-End Systems: <ul style="list-style-type: none"> • Meters-R-Us
Device Business Object: D1-SmartMeter	<ul style="list-style-type: none"> • Device Type: Electric Smart Meter - Scalar KWH • Serial Number: 54321 • Head-End System: Meters-R-Us • Status: Active
Device Configuration Business Object: D1-DeviceConfiguration	<ul style="list-style-type: none"> • Device Configuration Type: KWH - Scalar • Device: 54321 / Electric Smart Meter - Scalar KWH / Meters-R-Us / Active • Effective Date/Time: 01-01-2014 12:00:00AM • Time Zone: US Pacific Time • External ID: 12345
Measuring Component Business Object: D1-RegisterAutoRead	<ul style="list-style-type: none"> • Measuring Component Type: KWH - Auto Read • Device Configuration: Electric Smart Meter - Scalar KWH / Effective Date/Time: 01-01-2014 12:00:00AM / KWH - Scalar / 1 Measuring Component(s) / Active • How To Use: Additive • Channel ID: 2

Related Measuring Components Examples

These examples illustrate how to set up a primary/secondary relationship between channels (measuring components) on a single meter (device). This relationship is used with the Related Measuring Component Consumption Synchronization feature of Oracle Utilities Meter Data Management.

Scenario: Configure related interval channel and register measuring components such that the interval channel measuring component is a reference measuring component for the register measuring component. In this scenario, the register measuring component is considered the "primary" measuring component. In addition, the following also apply:

- Interval Data Size: 60 Minute
- Head-End System: Meters-R-Us (see above)

Object	Data and Field Values
Measuring Component Type Business Object: D1-IntervalChannelTypePhysical	<ul style="list-style-type: none"> • Measuring Component Type: KWH-60MIN • Description: KWH - 60 Minutes • Measuring Component Business Object: Interval Channel • Measurement Business Object: Measurement • Service Type: Electric • Consumptive / Subtractive: Consumptive • Seconds Per Interval: 01:00:00
Measuring Component Type Business Object: D1-AutoReadRegisterType	<ul style="list-style-type: none"> • Measuring Component Type: KWH-AUTO-READ • Description: KWH - Auto Read • Measuring Component Business Object: Register - Auto Read • Measurement Business Object: Measurement • Service Type: Electric • Consumptive / Subtractive: Subtractive • Read Method: Automatic
Device Configuration Type Business Object: D1-DeviceConfigurationType	<ul style="list-style-type: none"> • Device Configuration Type: KWH-INT-REG • Description: KWH - Two Channel (Interval and Scalar) • Device Configuration Business Object: Device Configuration • Service Type: Electric • Valid Measuring Component Types: <ul style="list-style-type: none"> • KWH - 60 Minutes • KWH - Auto Read

Object	Data and Field Values
Device Type Business Object: D1-SmartMeterType	<ul style="list-style-type: none"> • Device Type: ELEC-SMART-KWH • Description: Electric Smart Meter - KWH • Device Business Object: Smart Meter • Service Type: Electric • Head-End System (Fallback): Meters-R-Us • Device Classification: Meter • Valid Device Configuration Types: <ul style="list-style-type: none"> • KWH - Two Channel (Interval and Scalar) • Valid Head-End Systems: <ul style="list-style-type: none"> • Meters-R-Us
Device Business Object: D1-SmartMeter	<ul style="list-style-type: none"> • Device Type: Electric Smart Meter - KWH • Serial Number: 12345 • Head-End System: Meters-R-Us • Status: Active
Device Configuration Business Object: D1-DeviceConfiguration	<ul style="list-style-type: none"> • Device Configuration Type: KWH - Two Channel (Interval and Scalar) • Device: 12345 / Electric Smart Meter - KWH / Meters-R-Us / Active • Effective Date/Time: 01-01-2014 12:00:00AM • Time Zone: US Pacific Time • External ID: 54321
Primary Measuring Component Business Object: D1-RegisterAutoRead	<ul style="list-style-type: none"> • Measuring Component Type: KWH - Auto Read • Device Configuration: Electric Smart Meter - KWH / Effective Date/Time: 01-01-2014 12:00:00AM / KWH - Two Channel (Interval and Scalar) / 1 Measuring Component(s) / Active • Consumption Reference Measuring Component: N/A • How To Use: Additive • Channel ID: 2

Object	Data and Field Values
Secondary Measuring Component Business Object: D1-IntervalChannel	<ul style="list-style-type: none"> • Measuring Component Type: KWH - 60 Minutes • Device Configuration: Electric Smart Meter - KWH / Effective Date/Time: 01-01-2014 12:00:00AM / KWH - Two Channel (Interval and Scalar) / 1 Measuring Component(s) / Active • Consumption Reference Measuring Component: 12345 / 2 / KWH - Auto Read • How To Use: Additive • Channel ID: 1

Scenario: Configure related interval channel and register measuring components such that the register measuring component is a reference measuring component for the interval measuring component. In this scenario, the interval channel measuring component is considered the "primary" measuring component.

Note: This example uses the same measuring component types, device configuration type, device type, device, and device configuration as the previous example.

Object	Data and Field Values
Primary Measuring Component Business Object: D1-IntervalChannel	<ul style="list-style-type: none"> • Measuring Component Type: KWH - 60 Minutes • Device Configuration: Electric Smart Meter - KWH / Effective Date/Time: 01-01-2014 12:00:00AM / KWH - Two Channel (Interval and Scalar) / 1 Measuring Component(s) / Active • Consumption Reference Measuring Component: N/A • How To Use: Additive • Channel ID: 1
Secondary Measuring Component Business Object: D1-RegisterAutoRead	<ul style="list-style-type: none"> • Measuring Component Type: KWH - Auto Read • Device Configuration: Electric Smart Meter - KWH / Effective Date/Time: 01-01-2014 12:00:00AM / KWH - Two Channel (Interval and Scalar) / 1 Measuring Component(s) / Active • Consumption Reference Measuring Component: 12345 / 1 / KWH - 60 Minutes • How To Use: Additive • Channel ID: 2

Reader Remarks Example

Scenario: Create data to support the following types of reader remarks to be submitted with scalar register readings:

- Broken Seal - Can create a "Tamper" service issue monitor
- Tampering Detected

Object	Data and Field Values
Reader Remark Type - Broken Seal Business Object: D1-ReaderRemarkType	<ul style="list-style-type: none"> • Reader Remark Type: BROKEN_SEAL • Description: Broken Seal • Reader Remark BO: Reader Remark • Reader Remark Type Status: Active • Device Event Category: Reader Remark • Eligible for Processing: Yes • Service Issue Monitor Type: Tamper
Reader Remark Type - Tampering Business Object: D1-ReaderRemarkType	<ul style="list-style-type: none"> • Reader Remark Type: TAMPER • Description: Tampering Detected • Reader Remark BO: Reader Remark • Reader Remark Type Status: Active • Device Event Category: Reader Remark • Eligible for Processing: Yes • Service Issue Monitor Type:

Notes: Reader remark types are maintained using the Device Type portal. See Device Communication examples below for more information about service issue monitor types.

Manual Meter Example

Scenario: Create a scalar register manual meter such that register readings can be manually entered into the system.

Object	Data and Field Values
Measuring Component Type Business Object: D1-RegisterType	<ul style="list-style-type: none"> • Measuring Component Type: KWH-ELECTRIC • Description: Electric KWH • Measuring Component Business Object: Register • Measurement Business Object: Measurement • Service Type: Electric • Consumptive / Subtractive: Subtractive • Read Method: Manual Read

Object	Data and Field Values
Device Configuration Type Business Object: D1-DeviceConfigurationType	<ul style="list-style-type: none"> • Device Configuration Type: KWH-MANUAL • Description: KWH - Manual • Device Configuration Business Object: Device Configuration • Service Type: Electric • Valid Measuring Component Types: <ul style="list-style-type: none"> • Electric KWH
Device Type Business Object: D1-ManualMeterType	<ul style="list-style-type: none"> • Device Type: MANUAL_SCALAR_KWH • Description: Manual Meter - Scalar KWH • Device Business Object: Manual Meter • Service Type: Electric • Head-End System (Fallback): Meters-R-Us • Device Classification: Meter • Valid Device Configuration Types: <ul style="list-style-type: none"> • KWH - Manual
Device Business Object: D1-ManualMeter	<ul style="list-style-type: none"> • Device Type: Manual Meter - Scalar KWH • Serial Number: 54321 • Status: Active
Device Configuration Business Object: D1-DeviceConfiguration	<ul style="list-style-type: none"> • Device Configuration Type: KWH - Manual • Device: 54321 / Manual Meter - Scalar KWH / Meters-R-Us / Active • Effective Date/Time: 01-01-2014 12:00:00AM • Time Zone: US Pacific Time • External ID: 12345
Measuring Component Business Object: D1-Register	<ul style="list-style-type: none"> • Measuring Component Type: Electric KWH • Device Configuration: Manual Meter - Scalar KWH / Effective Date/Time: 01-01-2014 12:00:00AM / KWH - Manual / 1 Measuring Component(s) / Active • How To Use: Additive • Channel ID: 2

Note: To support import of usage for meters of this type, the device must be installed at a service point that references a measurement cycle that in turn references the appropriate head-end system. See Measurement Cycle examples below.

Item Examples

Scenario: Create a "badged" street light as follows:

- Serial Number is used as the unique identifier for items of this type.
- Consumption for items of this type is summarized.
- Average Daily KWH Consumption: 15 KWH.

Object	Data and Field Values
Device Configuration Type Business Object: D1-ItemConfigurationType	<ul style="list-style-type: none"> • Item Configuration Type: KWH-LAMP • Description: KWH - Street Lamp • Item Configuration Business Object: Device Configuration • Service Type: Electric
Device Type Business Object: D1-ItemType	<ul style="list-style-type: none"> • Device Type: STREET-LAMP-KWH • Description: Street Lamp - KWH • Item Business Object: Item • Device Classification: Item • Service Type: Electric • Consumption Source: Device Type Service Quantity • UOM: Kilowatt Hour • Device Type Service Quantity BO: Average Daily Estimated Item Consumption • Summarize Badged Items: • Valid Device Configuration Types: <ul style="list-style-type: none"> • KWH - Street Lamp
Device Type Service Quantity	<ul style="list-style-type: none"> • Item Type: Street Lamp - KWH • Effective Date: 01-01-2014 • Device Type Service Quantity: 15
Device Business Object: D1-Item	<ul style="list-style-type: none"> • Item Type: Street Lamp - KWH • Status: Active • Serial Number: 54321

Object	Data and Field Values
Device Configuration Business Object: D1-DeviceConfiguration	<ul style="list-style-type: none"> • Device Configuration Type: KWH - Street Lamp • Device: 54321 / Street Lamp - KWH / Install Date/Time: 01-01-2014 12:00:00AM / On / Active • Effective Date/Time: 01-01-2014 12:00:00AM • Time Zone: US Pacific Time • External ID: 12345 • Status: Active

Note: "Badged" item configurations are installed at service points with a service point category of "Item". See Service Point examples below.

Scenario: Create data for "unbadged" street lights as follows:

- Consumption for items of this type is summarized.
- Average Daily KWH Consumption: 15 KWH.

Object	Data and Field Values
Device Configuration Type Business Object: D1-ItemConfigurationType	<ul style="list-style-type: none"> • Item Configuration Type: KWH-LAMP • Description: KWH - Street Lamp • Item Configuration Business Object: Device Configuration • Service Type: Electric
Device Type Business Object: D1-ItemType	<ul style="list-style-type: none"> • Device Type: STREET-LAMP-KWH • Description: Street Lamp - KWH • Item Business Object: Item • Device Classification: Item • Service Type: Electric • Consumption Source: Device Type Service Quantity • UOM: Kilowatt Hour • Device Type Service Quantity BO: Average Daily Estimated Item Consumption • Summarize Badged Items: Yes • Valid Device Configuration Types: <ul style="list-style-type: none"> • KWH - Street Lamp
Device Type Service Quantity	<ul style="list-style-type: none"> • Item Type: Street Lamp - KWH • Effective Date: 01-01-2014 • Device Type Service Quantity: 15

Note: "Unbadged" items are installed at service points with a service point category of "Multi-Item". See Service Point examples below.

Service Points and Device Installation

This section provides configuration examples for setting up data related to device installation, including facilities and network locations, measurement cycles, service points, and install events.

Facilities and Network Location Example

Scenario: Create a transformer and network location that can be referenced by one or more service points, as follows:

- Feeder: 1234
- Substation: 4321
- Effective Date: 01-01-2014

Object	Data and Field Values
Facility Business Object: D1-Transformer	<ul style="list-style-type: none"> • Description: Transformer 1401 - 50KVA • Facility Type: Transformer • Service Type: Electric • Status: Active • External ID: 12345
Network Location Business Object: D1-ElectricityNetworkLocation	<ul style="list-style-type: none"> • Facility ID: Transformer 1401 - 50KVA / Electric / Active • State Date/Time: 01-01-2014 12:00:00AM • End Date/Time: • Feeder: 1234 • Substation: 4321 • External ID: 54321

Measurement Cycle Examples

Scenario: Create the data to support a meter reading cycle, as follows:

- Processed monthly
- Head-End System: Meters R Us (described above)
- This cycle will create a meter read download on the first day of the month (unless the first of the month falls on a weekend)

Object	Data and Field Values
Measurement Cycle Business Object: D1-MeasurementCycle	<ul style="list-style-type: none"> • Measurement Cycle: ELEC_RES_MON • Description: Electric Residential - Monthly • Measurement Cycle Schedule Business Object: Measurement Cycle Schedule • Measurement Cycle Route Business Object: Measurement Cycle Route

Object	Data and Field Values
Measurement Cycle Route Business Object: D1-MeasurementCycleRoute	<ul style="list-style-type: none"> • Measurement Cycle: Electric Residential - Monthly • Route: Route 1 • Description: Electric Residential Monthly - Route 1 • Service Provider: Meters R Us • Schedule Type: Meter Read Download
Measurement Cycle Schedule Business Object: D1-MeasurementCycleSchedule	<ul style="list-style-type: none"> • Measurement Cycle: Electric Residential - Monthly • Schedule Selection Date: 02-01-2014 (Saturday) • Schedule Selection Date: 02-03-2014

Service Point / Install Event Examples

Scenario: Create a residential electric service point at which a single meter can be installed and a corresponding install event, as follows:

- Service points of this type can be parent service points
- This service point is connected to the transformer and network location from the previous example
- This service point is the first service point on the measurement cycle and route from the previous example
- This service point is eligible for periodic estimation
- The manual meter described above is installed at this service point

Object	Data and Field Values
Service Point Type Business Object: D1-ServicePointType	<ul style="list-style-type: none"> • Service Point Type: ELEC_RES • Description: Electric Residential • Service Type: Electric • Parent Service Point: Usable as Parent SP • Service Point Business Object: Service Point • Service Point Category: Meter • Valid Device Types: <ul style="list-style-type: none"> • Device Type: Manual Meter - Scalar KWH

Object	Data and Field Values
Service Point Business Object: D1-ServicePoint	<ul style="list-style-type: none"> • Service Point Type: Electric Residential • Status: Active • Address: 1401 Flower Street, Glendale, CA, 99999, US • External Service Point ID: 54321 • Distribution Network Facility: Transformer 1401 - 50KVA / Electric / Active • Estimation Eligibility: Eligible • Measurement Cycle: <ul style="list-style-type: none"> • Measurement Cycle: Electric Residential - Monthly • Route: Route 1 • Sequence: 1
Install Event Business Object: D1-InstallEvent	<ul style="list-style-type: none"> • Device Configuration: Manual Meter - Scalar KWH / Effective Date/Time: 01-01-2014 12:00:00AM / KWH - Manual / 1 Measuring Component(s) / Active • Service Point: 1401 Flower Street, Glendale, CA, 99999, US / Electric Residential / Active • Status: On • Installation Date/Time: 01-01-2014 12:00:00AM • External ID: 54321 • Device On/Off Status: On • On/Off History: <ul style="list-style-type: none"> • Event Date/Time: 01-01-2014 12:00:00AM • Device On/Off Status: On

Note: For manual meters (those created from the D1-ManualMeter) business object, the relationship to a head-end system is facilitated via the meter's relationship to the service point (via an install event) which references a measurement cycle route that in turn references the head-end system.

Scenario: Create an electric service point at which a single "badged" item can be installed and a corresponding install event, as follows:

- Service points of this type cannot be parent service points
- The "badged" item described above is installed at this service point

Object	Data and Field Values
Service Point Type Business Object: D1-ServicePointType	<ul style="list-style-type: none"> • Service Point Type: STREET_LIGHT • Description: Street Light • Service Type: Electric • Parent Service Point: • Service Point Business Object: Service Point • Service Point Category: Item • Valid Device Types: <ul style="list-style-type: none"> • Device Type: Street Lamp - KWH
Service Point Business Object: D1-ServicePoint	<ul style="list-style-type: none"> • Service Point Type: Street Light • Status: Active • Address: 7 Winchester Street, Southborough, MA, 99999, US • External Service Point ID: 56789
Install Event Business Object: D1-ItemInstallEvent	<ul style="list-style-type: none"> • Device Configuration: Street Lamp - KWH / Effective Date/Time: 01-01-2014 12:00:00AM / KWH - Street Lamp / 0 Measuring Component(s) / Active • Service Point: 7 Winchester Street, Southborough, MA, 99999, US / Street Light / Active • Status: On • Installation Date/Time: 01-01-2014 12:00:00AM • External ID: 12345 • Device On/Off Status: On • On/Off History: <ul style="list-style-type: none"> • Event Date/Time: 01-01-2014 12:00:00AM • Device On/Off Status: On

Scenario: Create an electric service point at which a number of "unbadged" items can be installed, as follows:

- Service points of this type cannot be parent service points
- 5 "unbadged" street lights (described above) are installed at this point
- The street lights at this service point have an average daily consumption of 12 KWH (which differs from the average of 15)

Object	Data and Field Values
Service Point Type Business Object: D1-ServicePointType	<ul style="list-style-type: none"> • Service Point Type: STREET_LIGHT • Description: Street Light - Multiple • Service Type: Electric • Parent Service Point: • Service Point Business Object: Service Point • Service Point Category: Multi-Item • Valid Device Types: <ul style="list-style-type: none"> • Device Type: Street Lamp - KWH
Service Point Business Object: D1-ServicePoint	<ul style="list-style-type: none"> • Service Point Type: Street Light - Multiple • Status: Active • Address: 27 Park Street, Wakefield, MA 99999, US • External Service Point ID: 12345 • Multi-Item Information: <ul style="list-style-type: none"> • Start Date/Time: 01-01-2014 12:00:00AM • Installation Override Quantity (Daily): 12 • Multi-Item Count: <ul style="list-style-type: none"> • Item Type: Street Lamp - KWH • Count: 5

Note: "Unbadged" items do not require install events when installed at "multi-item" service points.

Device Communication

This section provides configuration examples for setting up data related to device communication, including device events and service investigative orders.

Device Event Example

Scenario: Create data to support the following types of device events to be received from a head-end system:

- Tampering
- Outage (Last Gasp)
- Power Restoration

Object	Data and Field Values
Device Event Type - Tampering Business Object: D1-StandardDeviceEventType	<ul style="list-style-type: none"> • Device Event Type: TAMPERING • Description: Tampering • Status: Active • Device Event BO: Device Event • Standard Event Name: Tamper • Device Event Category: Tamper Detection Events • Service Issue Monitor Type: Tamper
Device Event Type - Outage (Last Gasp) Business Object: D1-PairedEventFirstDvceType	<ul style="list-style-type: none"> • Device Event Type: OUTAGE-LASTGASP • Description: Outage - Last Gasp • Status: Active • Device Event BO: Device Event - Paired Event (First) • Standard Event Name: Last Gasp • Device Event Category: Power Status • Reporting Category: Power Status • Activity: Outage Activity
Device Event Type - Power Restoration Business Object: D1-PairedEventLastDvceType	<ul style="list-style-type: none"> • Device Event Type: RESTORE • Description: Power Restoration • Status: Active • Device Event BO: Device Event - Paired Event (Last) • Standard Event Name: Power On / Restore • Device Event Category: Power Status • Reporting Category: Power Status • Activity: Outage Activity

Service Investigative Order Example

Scenario: Create a service issue monitor type to create a service investigative order (and a custom "Field Activity") if a single instance of either of the following are received:

- Broken Seal (reader remark)
- Tamper (device event)

Object	Data and Field Values
Service Issue Monitor Type Business Object: D1-ServiceIssueMonitorType Service Task Business Object: Service Issue Monitor	<ul style="list-style-type: none"> • Service Task Type: TAMPERING • Description: Tamper • Service Task Class: Service Issue Monitor • Approval Required: Not Required • Evaluation Criteria: (broken seal) <ul style="list-style-type: none"> • Sequence: 10 • Evaluation Criteria Relationship: OR • Service Issue Monitor Evaluation Types: Device Event • Evaluation Details: <ul style="list-style-type: none"> • Device Event Type: Broken Seal • Number of Occurrences: 1 • Number of Days Back: 1 • Evaluation Criteria: (tamper) <ul style="list-style-type: none"> • Sequence: 20 • Evaluation Criteria Relationship: OR • Service Issue Monitor Evaluation Types: Device Event • Evaluation Details: <ul style="list-style-type: none"> • Device Event Type: Tamper • Number of Occurrences: 1 • Number of Days Back: 1 • Service Investigative Order: <ul style="list-style-type: none"> • Service Investigative Order Activity Type: Field Activity Type (custom) • Field Task Type: • Service Issue Monitor Discard Rules: <ul style="list-style-type: none"> • If Activity Found - Different Monitor Type: No • If Completed Activity Found: No

VEE Groups and Rules

This section provides examples of data configuration for VEE groups and rules, including:

- **VEE "Utility" Rule Examples**
- **VEE Validation Rule Examples**
- **VEE Estimation Rule Examples**

VEE "Utility" Rule Examples

This section provides examples of VEE rules that can be used when configuring VEE groups and rules with Oracle Utilities Meter Data Management (referred to as "utility" rules).

Exception Handler

Exception Handler rules are used to define options and logic to terminate the VEE process when a set of user configured criteria are met.

Scenario: Configure a VEE rule that terminates VEE processing if/when five "Insufficient Input Data " exceptions are encountered and creates an "Exception Termination" exception.

Object	Data and Field Values
VEE Rule Business Object: D1- VEERuleExceptionHandler	<ul style="list-style-type: none"> • VEE Group: KWH_INT_INITIAL_LOAD • VEE Rule: EXCEPTION_HNDLR • Sequence: 50 • Description: Exception Handler • Category: Validation Rules • Start Date: 01-01-2014 • Comparison Results: <ul style="list-style-type: none"> • Exception Type: Insufficient Input Data • Number of Exceptions: 5 • Exception Criteria Relationship: And • Exception Handler: <ul style="list-style-type: none"> • Exception Type: Exception Termination • Override To Do Type: • Override To Do Role:

Execute VEE Group

Execute VEE Group rules are used to execute a specified VEE group. This allows rules that are used frequently to be bunched under a single VEE group, which can be referenced/called by other rules as needed.

Scenario: Configure a VEE rule that executes a VEE group that contains a set of "common" rules in validating incoming initial measurements, as follows:

- "Common" rules include a duplicate initial measurement check, multiplier check, and unit of measure check

Object	Data and Field Values
VEE Group Business Object: D1-VEEGroup	<ul style="list-style-type: none"> • VEE Group: COMMON_VAL • Description: Common Validations • VEE Rules List (zone): <ul style="list-style-type: none"> • Duplicate IMD Exists • Multiplier Check • Unit of Measure Check
VEE Rule Business Object: D1-VEERuleReferredGroup	<ul style="list-style-type: none"> • VEE Group: KWH_INT_INITIAL_LOAD • VEE Rule: INITIAL_LOAD_COMMON • Sequence: 10 • Description: Initial Load - Common Rules • Category: Validation Rules • Start Date: 01-01-2014 • Referenced VEE Group: <ul style="list-style-type: none"> • VEE Group: COMMON_VAL • Insufficient Input Data Exception: <ul style="list-style-type: none"> • Exception Type: No Input Data • Exception Severity: Information

VEE Group Matrix (Factor)

VEE Group Matrix (Factor) rules are used to define a reference to a factor (of type VEE Group) where the values of the factor are a list of VEE groups. This allows creating a VEE rule that can select from a list of VEE groups (referred to as a matrix) whose rules to execute next.

Scenario: Configure a VEE rule that executes a specific VEE group based on the location in which the service point is located (Oakland or San Francisco)..

Object	Data and Field Values
Characteristic Type	<ul style="list-style-type: none"> • Characteristic Type: CA-LOC • Description: CA Locations • Type of Char Value: Predefined Value • Characteristic Values: <ul style="list-style-type: none"> • Oakland <ul style="list-style-type: none"> • Characteristic Value: OK • Description: Oakland • San Francisco <ul style="list-style-type: none"> • Characteristic Value: SF • Description: San Francisco
VEE Group (High-Low Check for Oakland) Business Object: D1-VEEGroup	<ul style="list-style-type: none"> • VEE Group: HIGH-LOW-OK • Description: High-Low Check for Oakland • VEE Rules List (zone): <ul style="list-style-type: none"> • High / Low Check for Oakland
VEE Group (High-Low Check for San Francisco) Business Object: D1-VEEGroup	<ul style="list-style-type: none"> • VEE Group: HIGH-LOW-OK • Description: High-Low Check for San Francisco • VEE Rules List (zone): <ul style="list-style-type: none"> • High / Low Check for San Francisco
Factor Business Object: D1-FactorVEEGroup	<ul style="list-style-type: none"> • Factor: HIGH-LOW-BY-LOCATION • Description: High/Low by Location Factor • Factor Class: VEE Group • Characteristic Source Algorithm: Factor Characteristic Source Service Point • Factor Characteristic Type: CA Locations
Factor Value (Oakland) Business Object: D1-FactorValueVEEGroup	<ul style="list-style-type: none"> • Factor: High/Low by Location Factor • Factor Characteristic Type: CA Locations • Factor Characteristic Value: OK • Effective Date/Time: 01-01-2014 12:00:00AM • VEE Group: HIGH-LOW-OK

Object	Data and Field Values
Factor Value (San Francisco) Business Object: D1- FactorValueVEEGroup	<ul style="list-style-type: none"> • Factor: High/Low by Location Factor • Factor Characteristic Type: CA Locations • Factor Characteristic Value: SF • Effective Date/Time: 01-01-2014 12:00:00AM • VEE Group: HIGH-LOW-SF
VEE Rule Business Object: D1- VEERuleGroupFactor	<ul style="list-style-type: none"> • VEE Group: KWH_INT_INITIAL_LOAD • VEE Rule: HIGHLOW_BY_CUST_AREA • Sequence: 10 • Description: High / Low by Customer Area • Category: Validation Rules • Start Date: 01-01-2014 • VEE Group Matrix (Factor): <ul style="list-style-type: none"> • VEE Group Matrix (Factor): High/Low by Location Factor • If Group Not Found: Skip • Insufficient Input Data Exception: <ul style="list-style-type: none"> • Exception Type: Insufficient Input Data • Exception Severity: Issues

Successful Termination

Successful Termination rules are used to define options and logic to successfully terminate VEE processing for any initial measurement that passes a pre-defined set of validations before accumulating a pre-defined number of exceptions.

Scenario: Configure a VEE rule that successfully terminates VEE processing if one or fewer "Insufficient Input Data " exceptions are encountered.

Object	Data and Field Values
VEE Rule Business Object: D1- VEERuleSuccessfulTermination	<ul style="list-style-type: none"> • VEE Group: KWH_INT_INITIAL_LOAD • VEE Rule: SUCCESSFUL_TERMINATION • Sequence: 90 • Description: Successful Termination • Category: Validation Rules • Start Date: 01-01-2014 • Comparison Results: <ul style="list-style-type: none"> • Exception Type: Insufficient Input Data • Number of Exceptions: 1 • Exception Criteria Relationship: Or

VEE Validation Rule Examples

This section provides examples of the VEE validation rules provided with the base package.

Duplicate IMD Check

Business Object: D1-DuplicateIMDCheck

Scenario: Configure a VEE rule that creates an exception if the current initial measurement is determined to be a duplicate of an existing “Final” initial measurement for the same measuring component.

- **VEE Group:** Initial Load Validations
- **VEE Rule:** DUPLICATE_CHECK
- **Sequence:** 10
- **Description:** Duplicate IMD Check
- **Category:** Validation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Duplicate IMD Exception Configuration:**
 - **Exception Type:** Duplicate IMD Detected
 - **Exception Severity:** Terminate
 - **Override To Do Type:** N/A
 - **Override To Do Role:** N/A

Ensure IMD Exists for Sibling MCs

Business Object: D2-EnsureIMDExistsForSibling

Scenario: Configure a VEE rule that creates an exception if initial measurements are not found for all sibling measuring components of the current initial measurement's measuring component.

- **VEE Group:** Initial Load Validations
- **VEE Rule:** CHECK_SIBLING_MC_IMD
- **Sequence:** 10
- **Description:** Check IMDs for Sibling MCs
- **Category:** Validation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Issues
- **IMD Not Found Exception:**
 - **Exception Type:** No Data Found
 - **Exception Severity:** Issues

Final Measurement Validation

Business Object: D2-FinalMeasurementValidation

Scenario: Configure a VEE rule that creates an exception if an initial measurement has lower quality condition codes than existing final measurements for the same time period as the initial measurement.

- **VEE Group:** Initial Load Validations
- **VEE Rule:** FINAL_MEASUREMENT_VALIDATION
- **Sequence:** 10
- **Description:** Final Measurement Validation
- **Category:** Validation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Measurement Replacement Validation:**
 - **Replacement Condition Code:** No Read — System
- **Measurement Replacement Exception:**
 - **Exception Type:** Measurement Replacement Exception
 - **Exception Severity:** Issues

High/Low Check

Business Object: D2-VEERuleHighLowCheck

Scenario: Configure a VEE rule that performs both a High check and Low check against historical data. When gathering historical data for comparison, the rule will use up to 5% user-edited data, up to 20% system estimated data, and up to 10% non-normal data. In addition, if consumption for the current initial measurement is deemed to be “low”, the rule will also check for outages.

- **VEE Group:** Initial Load Validations
- **VEE Rule:** HIGH-LOW-CHECK
- **Sequence:** 10
- **Description:** High/Low Check
- **Category:** Validation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **High Low Check:**
 - **High Low Check:** Both
 - **High/Low Limits from master config:** No
 - **High Tolerance (%):** 200
 - **Low Tolerance (%):** 49
 - **Required Historical Data (%):** 20
 - **Historical Pre-Window:** 10
 - **Historical Post-Window:** 0
 - **Allowed Historical User Edited Data (%):** 5

- **Allowed Historical System Estimated Data (%)**: 20
- **Allowed Historical Non Normal Data (%)**: 10
- **Historical First Look**: Last Reading
- **Comparison Method**: Average
- **Perform Outage Check**: Perform If Low Consumption
- **Outage Activity Type**: Outage Activity
- **Insufficient Input Data Exception**:
 - **Exception Type**: Insufficient Input Data
 - **Exception Severity**: Issues
- **No Historical Data Found Exception**:
 - **Exception Type**: No Data Found
 - **Exception Severity**: Issues
- **High Check Exception**:
 - **Exception Type**: Consumption Exceeds Threshold
 - **Exception Severity**: Issues
- **Low Check Exception**:
 - **Exception Type**: Consumption Less Than Threshold
 - **Exception Severity**: Issues
- **Low Check Outage Exception**:
 - **Exception Type**: Consumption Less Than Threshold due to Outage
 - **Exception Severity**: Issues

Interval Replacement Rule

Business Object: D2-IntervalReplacementRule

Scenario: Configure a VEE rule that determines whether or not an interval reading would replace any existing measurements and if so, will reject the incoming measurements.

- **VEE Group**: Interval Validations
- **VEE Rule**: INTERVAL_REPLACEMENT
- **Sequence**: 10
- **Description**: Interval Replacement Rule
- **Category**: Validation Rules.
- **Start Date**: 05-01-2010
- **End Date**: N/A
- **Replacement Handling Method**: Reject All
- **Insufficient Input Data Exception**:
 - **Exception Type**: Insufficient Input Data
 - **Exception Severity**: Issues
- **Interval Replacement Exception**:
 - **Exception Type**: Interval Data Replacement

- **Exception Severity:** Issues

Interval Size Validation

Business Object: D2-IntervalSizeValidation

Scenario: Configure a VEE rule that issues an exception if the interval size of the incoming initial measurement does not match the value defined for its measuring component type.

- **VEE Group:** Interval Validations
- **VEE Rule:** SPI_CHECK
- **Sequence:** 10
- **Description:** SPI Check
- **Category:** Validation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Issues
- **Invalid Interval Size Exception:**
 - **Exception Type:** Interval Size Discrepancy
 - **Exception Severity:** Issues

Interval Spike Check

Business Object: D2-IntervalSpikeCheck

Scenario: Configure a VEE rule that checks the entire measurement for interval spikes, and flags those interval values identified as spikes, based on a spike tolerance of 70%.

- **VEE Group:** Interval Validations
- **VEE Rule:** SPIKE_CHECK
- **Sequence:** 10
- **Description:** Interval Spike Check
- **Category:** Validation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Minimum Number of Intervals:** 8
- **Spike Check Method:** Entire Cut
- **Spike Tolerance (%):** 70
- **Condition Flag Value to Interval Spike(s):** Spike - Treat As Missing
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Issues
- **Interval Spike Check Exception:**
 - **Exception Type:** Interval Check Detected
 - **Exception Severity:** Issues

Multiplier Check

Business Object: D2-RegisterMultiplierCheck

Scenario: Configure a VEE rule that checks if the meter multiplier supplied in the initial measurement differs from the meter multiplier defined for the measuring component.

- **VEE Group:** Initial Load Validations
- **VEE Rule:** MULTIPLIER_CHECK
- **Sequence:** 10
- **Description:** Multiplier Check
- **Category:** Validation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Issues
- **Multiplier Exception:**
 - **Exception Type:** Device Multiplier Discrepancy
 - **Exception Severity:** Issues

Negative Consumption Check

Business Object: D2-NegativeConsumptionCheck

Scenario: Configure a VEE rule that checks for negative consumption and creates an exception if negative consumption is detected.

- **VEE Group:** Initial Load Validations
- **VEE Rule:** NEGATIVE_CONSUMPTION_CHECK
- **Sequence:** 10
- **Description:** Negative Consumption Check
- **Category:** Validation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Issues
- **Negative Consumption Exception:**
 - **Exception Type:** Unexpected Negative Consumption
 - **Exception Severity:** Issues

Raise Missing Quantity Exception

Business Object: D2-RaiseMissingQuantityExcp

Scenario: Configure a VEE rule that creates an exception if more than 10 percent of the interval values within an initial measurement are missing.

- **VEE Group:** Interval Validations
- **VEE Rule:** MISSING_EXCEPTION
- **Sequence:** 10
- **Description:** Excessive Gaps
- **Category:** Validation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Missing Quantity Data:**
 - **Percentage Threshold of Missing Intervals:** 10
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Issues
- **Missing Quantity Exception:**
 - **Exception Type:** Gaps Detected in Data
 - **Exception Severity:** Issues

Scalar Replacement Rule

Business Object: D2-ScalarReplacementRule

Scenario: Configure a VEE rule that determines whether or not a scalar reading will replace existing measurements and if so, whether it will reject the incoming measurements if the existing measurements were manually edited.

- **VEE Group:** Scalar Validations
- **VEE Rule:** SCALAR_REPLACEMENT
- **Sequence:** 10
- **Description:** Scalar Replacement
- **Category:** Validation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Replacement Handling Method:** Reject If Existing Data Is Manually Edited
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Issues
- **Scalar Replacement Exception:**
 - **Exception Type:** Scalar Data Replacement
 - **Exception Severity:** Issues

Sum Check

Business Object: D2-SumCheck

Scenario: Configure a VEE rule that creates an exception if the percentage difference between the consumption of the incoming initial measurement and consumption for its related measuring components is more than 95%.

The following VEE rule creates an exception if the percentage difference between the consumption of the incoming initial

measurement and consumption for its related measuring components is more than 95%.

- **VEE Group:** Interval Validations
- **VEE Rule:** SUM_CHECK
- **Sequence:** 10
- **Description:** Sum Check
- **Category:** Validation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Sum Check:**
 - **Percentage Tolerance (%):** 95
 - **Tolerance:** N/A
 - **Meter Multiplier Tolerance:** N/A
 - **Required Related MC Measurements (%):**
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Issues
- **Sum Check Exception:**
 - **Exception Type:** Consumption Differs from Reference Amount
 - **Exception Severity:** Issues
- **Insufficient Related MC Measurements Exception:**
 - **Exception Type:**
 - **Exception Severity:**
- **No Data Found Exception:**
 - **Exception Type:** No Data Found
 - **Exception Severity:** Issues

Unit Of Measure Check

Business Object: D2-UOMCheck

Scenario: Configure a VEE rule that creates an exception if the unit of measure specified in the initial measurement does not match the unit of measure defined for its measuring components' type.

- **VEE Group:** Initial Load Validations
- **VEE Rule:** UOM_CHECK
- **Sequence:** 10
- **Description:** Unit of Measure Check
- **Category:** Validation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Issues
- **Unit of Measure Exception:**
 - **Exception Type:** Unit of Measure Discrepancy
 - **Exception Severity:** Issues

Zero Consumption Check

Business Object: D2-ZeroConsumptionCheck

Scenario: Configure a VEE rule that creates an exception if the total consumption for the incoming initial measurement is zero. If the consumption is zero, this rule will also check for outage activities.

- **VEE Group:** Initial Load Validations
- **VEE Rule:** ZERO_CONSUMPTION_CHECK
- **Sequence:** 10
- **Description:** Zero Consumption Check
- **Category:** Validation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Perform Outage Check If Zero Consumption:** Yes
- **Outage Activity Type:** Outage Activity
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Issues
- **Zero Consumption:**
 - **Exception Type:** Zero Consumption
 - **Exception Severity:** Issues
- **Zero Consumption Outage:**
 - **Exception Type:** Zero Consumption Due to Outage

- **Exception Severity:** Issues

VEE Estimation Rule Examples

This section provides examples of the VEE estimation rules provided with the base package.

Interval Adjustment From Scalar

Business Object: D2-IntervalAdjustmentFrmScalar

Scenario: Configure a VEE rule that adjusts all interval values in an interval measurement based on final measurements for a related scalar measuring component, and flags all adjusted intervals as “system estimate”.

- **VEE Group:** Interval Estimations
- **VEE Rule:** INTERVAL_ADJUST
- **Sequence:** 10
- **Description:** Interval Adjustment From Scalar
- **Category:** Estimation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Interval Adjustment From Scalar:**
 - **Interval to Adjust:** All
 - **Bottom Range Condition Value:**
 - **Top Range Condition Value:**
 - **Interpolate Zero Quantity Intervals:**
 - **Maximum Consecutive Hours to Interpolate:**
 - **Condition Value for Adjusted Intervals:** System Estimate
 - **Adjust All Intervals for Empty IMD:**
 - **Required Scalar MC Measurement (%):**
- **Override Condition for High Quality Source Measurements:**
 - **Minimum Condition Quality to Override:**
 - **Override Condition Value:**
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Information
- **Interval Adjustment Exception:**
 - **Exception Type:** Interval Measurement Adjusted Per Scalar Value
 - **Exception Severity:** Information
- **Insufficient Related MC Measurements Exception:**
 - **Exception Type:**
 - **Exception Severity:**
- **Interpolization Failure Exception:**
 - **Exception Type:**

- **Exception Severity:**

Interval Averaging Estimation

Business Object: D2-IntervalAveragingEstimation

Scenario: Configure a VEE rule that estimates interval values in an initial measurement based on averaging of three historical interval values providing that no more than 10% of the interval values in the measurement are missing.

- **VEE Group:** Interval Estimations
- **VEE Rule:** INTERVAL_AVERAGING_ESTIMATION
- **Sequence:** 10
- **Description:** Interval Averaging Estimation
- **Category:** Estimation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Interval Averaging Estimation:**
 - **Work Calendar:** US Work Calendar 1
 - **Maximum Percentage Missing Intervals:** 10
 - **Estimate if Not Attached to SP:** Estimate
 - **Include User Edited Intervals:** Include
 - **Include Intervalized Scalar Data:** No
 - **Same Day of Week Scan Range:** 3
 - **Neighboring Day Scan Range:** 3
 - **Sunday Scan Range:** 3
 - **Holiday Scan Range:** 2
 - **Intervals to Average:** 3
 - **Condition Value Estimates Created:** System Estimate
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Information
- **Max Percentage Missing Intervals Exception:**
 - **Exception Type:** Maximum Percentage of Missing Data Exceeded
 - **Exception Severity:** Information

Interval Interpolation Estimation

Business Object: D2-IntervalInterpolationEst

Scenario: Configure a VEE rule that estimates up to three hours of missing interval values using linear interpolation providing that no more than 20% of the interval values in the measurement are missing.

- **VEE Group:** Interval Estimations
- **VEE Rule:** INTERPOLATE_GAPS
- **Sequence:** 10

- **Description:** Interpolate Gaps
- **Category:** Estimation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Interval Interpolation Estimation:**
 - **Maximum Percentage Missing Intervals:** 20
 - **Estimate if Not Attached to SP:** Estimate
 - **Maximum Hours to Interpolate:** 3
 - **Condition Value for Estimates Created:** System Estimate
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Information
- **Maximum Hours Exceeded Exception:**
 - **Exception Type:** Maximum Gaps Size for Estimation Exceeded
 - **Exception Severity:** Information
- **Max Percentage Missing Intervals Exception:**
 - **Exception Type:** Maximum Percentage of Missing Data Exceeded
 - **Exception Severity:** Information

Interval Profile Estimation

Business Object: D2-IntervalProfileEstimation

Scenario: Configure a VEE rule that estimates interval measurements based on the “kWh - Profile:” profile measuring component (defined as a factor) provided that no more than 20% of the interval values in the measurement are missing.

- **VEE Group:** Interval Estimations
- **VEE Rule:** INTERVAL_PROFILE_EST
- **Sequence:** 10
- **Description:** Interval Profile Estimation
- **Category:** Estimation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Interval Profile Estimation:**
 - **Maximum Percentage Missing Intervals:** 20
 - **Estimate if Not Attached to SP:** Estimate
 - **Factor:** kWh - Profile
 - **Condition Value for Estimates Created:** System Estimate
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Information
- **Max Percentage Missing Intervals Exception:**

- **Exception Type:** Maximum Percentage of Missing Data Exceeded
- **Exception Severity:** Information

Scalar Calculation From Interval

Business Object: D2-ScalarCalcFromInterval

Scenario: Configure a VEE rule that calculates a scalar measurement value based on the total consumption for the same date/time range for a related interval measuring component.

- **VEE Group:** Scalar Estimations
- **VEE Rule:** SCALAR_CALC_FROM_INTERVAL
- **Sequence:** 10
- **Description:** Scalar Calculation From Interval
- **Category:** Estimation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Scalar Calculation From Interval:**
 - **IMD Created Condition Value:** System Estimate
 - **Required Scalar MC Measurement (%):** 90
- **Override Condition for High Quality Source Measurements:**
 - **Minimum Condition Quality to Override:**
 - **Override Condition Value:**
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Information
- **Insufficient Related MC Measurements Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Information

Scalar Estimation

Business Object: D2-ScalarEstimation

Scenario: Configure a VEE rule that estimates scalar usage based on measurements from one year-ago, providing that no more than 20% of the historical data was either user-edited to estimated.

- **VEE Group:** Scalar Estimations
- **VEE Rule:** SCALAR_ESTIMATION
- **Sequence:** 10
- **Description:**
- **Category:** Estimation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Scalar Estimation:**
 - **Historical Percentage Required:** 50

- **Historical First Look:** Last Year
- **Historical Pre-Window:** 10
- **Historical Post-Window:** 5
- **Allowed Historical System Estimated Data (%):** 20
- **Allowed Historical User Edited Data (%):** 20
- **Interim Reading High Threshold Percentage:** 50
- **Interim Reading Low Threshold Percentage:** 50
- **Condition Value Estimates Created:** System Estimate
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Information
- **Estimated Value Exceeds Limit Exception:**
 - **Exception Type:** Consumption Differs from Reference Amount
 - **Exception Severity:** Information

Scalar Profile Estimation

Business Object: D2-ScalarProfileEstimation

Scenario: Configure a VEE rule that estimates scalar measurement values based on a specified profile measuring component, providing that no more than 20% of the historical data was estimated.

VEE Group: s

VEE Rule: SCALAR_PROFILE_EST

Sequence: 10

Description: Scalar Profile Estimation

Category: Estimation Rules.

Start Date: 05-01-2010

End Date: N/A

Scalar Profile Estimation:

- **Historical Percentage Required:** 50
- **Allowed Historical System Estimated Data (%):** 20
- **Estimation Profile Factor:** kWh - Profile
- **Interim Reading High Threshold Percentage:** 50
- **Interim Reading Low Threshold Percentage:** 50
- **Condition Value for Estimates Created:** System Estimate

Insufficient Input Data Exception:

- **Exception Type:** Insufficient Input Data
- **Exception Severity:** Information

Estimated Value Exceeds Limit Exception:

- **Exception Type:** Consumption Differs from Reference Amount
- **Exception Severity:** Information

Scalar Proration

Business Object: D2-ScalarProfileEstimation

Scenario: Configure a VEE rule that prorates a scalar measurement value based on the preceding and subsequent measurements.

- **VEE Group:** Scalar Estimations
- **VEE Rule:** SCALAR_PRORATION
- **Sequence:** 10
- **Description:** Scalar Proration
- **Category:** Estimation Rules.
- **Start Date:** 05-01-2010
- **End Date:** N/A
- **Scalar Proration:**
 - **Minimum Condition to Preserve Measurement:**
 - **IMD Created Condition Value:** External Estimate
 - **Minimum Preceding Read Condition Quality:** Regular
 - **Minimum Subsequent Read Condition Quality:** Regular
- **Override Condition for High Quality Source Measurements:**
 - **Minimum Condition Quality to Override:** Regular
 - **Override Condition Value:** System Estimate
- **Insufficient Input Data Exception:**
 - **Exception Type:** Insufficient Input Data
 - **Exception Severity:** Information
- **Reading Quality Exception:**
 - **Exception Type:** Reading Quality Exception
 - **Exception Severity:** Issues

Usage Groups and Rules

This section provides examples of data configuration for usage groups and rules, including:

- **Usage "Utility" Rule Examples**
- **Usage Calculation Rule Examples**

Usage "Utility" Rule Examples

This section provides examples of usage rules that can be used when configuring usage groups and rules with Oracle Utilities Meter Data Management (referred to as "utility" rules).

Execute Usage Group

Execute Usage Group rules are used to define business logic to allow reference to a usage group. This allows rules that are used frequently to be bunched under a single usage group, which can be referenced/called by other rules as needed.

Scenario: Configure a usage rule that executes a usage group that contains a rule that multiplies interval consumption by a loss factor.

Object	Data and Field Values
Usage Group Business Object: D1-UsageGroup	<ul style="list-style-type: none"> • Usage Group: APPLY_MATH_GROUP • Description: Apply Math to Interval Data • Usage Rules List (zone): <ul style="list-style-type: none"> • Apply Math to Interval Data
Usage Rule Business Object: D1-UsageRuleReferredGroup	<ul style="list-style-type: none"> • Usage Group: GET_INTERVAL_DATA • Usage Rule: APPLY_MATH_INTERVAL_DATA • Sequence: 10 • Description: Apply Math - Interval Data • Usage Rule Category: Usage Calculation • Referenced Usage Group: <ul style="list-style-type: none"> • Usage Group: APPLY_MATH_GROUP

Usage Calculation Rule Examples

This section provides examples of the usage calculation rules provided with the base package.

Apply Math (Interval Data)

Business Object: D2-ApplyMathInt

Scenario: Configure a usage rule that calculates usage based on an interval measurement by calculating the total of the interval values for measuring components that measure KWH.

- **Usage Group:** Interval Usage
- **Usage Rule:** TOTAL_KWH
- **Sequence:** 10
- **Description:** Total KWH
- **Category:** Usage Calculation
- **Result**

- **UOM:** Kilowatt-Hours
- **TOU:**
- **SQI:**
- **Calculation Details:**
 - **Calculation Type:** Single Value
 - **Single Value Variable:** V1
- **Variables:**
 - **V1:**
 - **Variable Type:** Channel Accumulation
 - **UOM:** Kilowatt-Hours
 - **Set Function:** Sum

Get Interval Data

Business Object: D2-GetIntervalData

Scenario: Configure a usage rule that calculates the sum of the interval values in interval measurements that record kilowatt hours, and extracts a snapshot of the resulting interval data.

- **Usage Group:** Interval Usage
- **Usage Rule:** GET_INTERVAL_KWH DATA
- **Sequence:** 10
- **Description:** Get Interval Data - KWH
- **Category:** Usage Calculation
- **Interval Data Details:**
 - **UOM:** Kilowatt-Hours
 - **TOU:**
 - **SQI:**
 - **Calculation Function:** Sum
 - **Extract Interval Data:** Yes

Get Item Counts and Consumption

Business Object: D2-GetItemCountsConsumption

Scenario: Configure a usage rule that calculates consumption for service points where street lights are installed.

- **Usage Group:** Street Light Consumption
- **Usage Rule:** GET_STREET_LIGHT_COUNT AND CONSUMPTION
- **Sequence:** 10
- **Description:** Get Street Light Count and Consumption
- **Category:** Usage Calculation
- **Item Processing Configuration:**
 - **Restrict to Items of This Type:**
 - **Item Types:**
 - **Item Type:** Street Light — 10 KWH

- **SQI:**

Get Scalar Details

Business Object: D2-GetScalar

Scenario: Configure a usage rule that retrieves kilowatt hours from scalar measurements and adds the results to the Service Quantity list in the usage transaction.

- **Usage Group:** Scalar Usage
- **Usage Rule:** GET_SCALAR_KWH
- **Sequence:** 10
- **Description:** Get Scalar KWH
- **Category:** Usage Calculation
- **Scalar Details:**
 - **UOM:** Kilowatt-Hours
 - **TOU:**
 - **SQI:**
 - **Build Service Quantity:** Yes (checked)

Get TOU Mapped Usage

Business Object: D2-GetTOUUsage

Scenario: Configure a usage rule that calculates TOU values from interval measurements that record kilowatt hours based on a specified TOU map.

- **Usage Group:** Interval Usage
- **Usage Rule:** GET_TOU_MAPPED_KWH
- **Sequence:** 10
- **Description:** Get TOU Mapped KWH Usage
- **Category:** Usage Calculation
- **TOU Mapping Details:**
 - **Result SQI:**
 - **Unit of Measure:** Kilowatt-Hours
 - **Service Quantity Identifier:**
 - **Time of Use Calculate Function:** Sum
 - **TOU Map:** Summer / Winter, 15 minute interval

Profile Accumulation

Business Object: D2-ProfileAccumulation

Scenario: Configure a usage rule that accumulates KWH profile factors and the customer's KWH usage, and derives TOU values from the result based on a specified TOU map.

- **Usage Group:** Rate Comparison
- **Usage Rule:** KWH_PROFILE_ACCUMULATION
- **Sequence:** 10
- **Description:** KWH Profile Accumulation
- **Category:** Usage Calculation

- **Vector Source Configuration:**
 - **Vector Type:** Channels Linked to Usage Subscription
 - **Unit of Measure:** Kilowatt-Hours
 - **Time of Use:**
 - **Service Quantity Identifier:**
 - **Target Interval Size:** 01:00:00
- **Result Processing Configuration:**
 - **Apply TOU Map to Derived Vector:** Yes
 - **TOU Map:** Summer / Winter, 15 minute interval
- **Result Storage Configuration:**
 - **Insert Primary SQ Entry:** Yes
 - **Save Derived Vector:** Yes
 - **Service Quantity Identifier:**
 - **Extract Interval Data:** Yes

Vector and Service Quantity Math

Business Object: D2-Math

Scenario: Configure a usage rule that calculates KVA (kilovolt amperes) from KW (kilowatts) and KVAR (kilovolt amperes reactive).

- **Usage Group:** Interval Usage
- **Usage Rule:** CALC_KVA
- **Sequence:** 10
- **Description:** Calculate KVA from KW and KVAR
- **Category:** Usage Calculation
- **Vector 1:** (IV1)
 - **Type:** Channels Linked To Usage Subscription
 - **Unit of Measure:** Kilowatts
 - **Time of Use:**
 - **Service Quantity Identifier:**
 - **Target Unit of Measure:** Kilowatts
- **Vector 2:** (IV2)
 - **Type:** Channels Linked To Usage Subscription
 - **Unit of Measure:** Kilovolt-Amperes Reactive
 - **Time of Use:**
 - **Service Quantity Identifier:**
 - **Target Unit of Measure:** Kilovolt-Amperes Reactive
- **Scalar Variables:** N/A
- **Vector Processing:**
 - **Common Interval Size:** 01:00:00

- **Vector Formula Source:** Simple Vector Formula
- **Simple Vector Formula:** (IV1=IV2)
- **Result:**
 - **Unit of Measure:** Kilovolt-Amperes
 - **Service Quantity Identifier:**
 - **Insert SQ Entry:** Yes
 - **SQ Entry Quantity Source:** Set Function Against Derived Vector
 - **Set Function Against Derived Vector:** Max
 - **Save Derived Vector:** Yes
 - **Extract Interval Data:** No
 - **Apply TOU Map to Derived Vector:** No

Scenario: Configure a usage rule that calculates net usage for customers with solar panels by subtracting generated usage from consumed usage, and multiplying the resulting interval values by a "power factor" stored as a factor. This rule also extracts a snapshot of the interval data resulting from the rule.

- **Usage Group:** Interval Usage
- **Usage Rule:** CALC_NET_USAGE_PF
- **Sequence:** 10
- **Description:** Calculate Net Usage with Power Factor Applied
- **Category:** Usage Calculation
- **Vector 1:** (IV1)
 - **Type:** Channels Linked To Usage Subscription
 - **Unit of Measure:** Kilowatt-Hours
 - **Time of Use:**
 - **Service Quantity Identifier:** Consumed
 - **Target Unit of Measure:** Kilowatt-Hours
- **Vector 2:** (IV2)
 - **Type:** Channels Linked To Usage Subscription
 - **Unit of Measure:** Kilowatt-Hours
 - **Time of Use:**
 - **Service Quantity Identifier:** Generated
 - **Target Unit of Measure:** Kilowatt-Hours
- **Scalar Variables:**
 - **V1:**
 - **Sequence:** 1
 - **Factor:** Power Factors
- **Vector Processing:**
 - **Common Interval Size:** 01:00:00
 - **Vector Formula Source:** Simple Vector Formula

- **Simple Vector Formula:** $(IV1=IV2)*V1$
- **Result:**
 - **Unit of Measure:** Kilowatt-Hours
 - **Service Quantity Identifier:** Net Consumption
 - **Insert SQ Entry:** Yes
 - **SQ Entry Quantity Source:** Set Function Against Derived Vector
 - **Set Function Against Derived Vector:** Total
 - **Save Derived Vector:** Yes
 - **Extract Interval Data:** Yes
 - **Apply TOU Map to Derived Vector:** Yes
 - **TOU Map:** Year round schedule, 15 minute interval
 - **Time of Use Calculate Function:** Sum

Validate Against Tolerance\

Business Object: D2-ValAgainstTol

Scenario: Configure a usage rule that validates that the sum of calculated usage values for kilowatt hours is less than or equal to a specified tolerance.

- **Usage Group:** Interval Usage
- **Usage Rule:** VALIDATE_AGAINST_TOLERANCE
- **Sequence:** 10
- **Description:** Validate Against Tolerance
- **Category:** Usage Calculation
- **Validation Details:**
 - **UOM:** Kilowatt-Hours
 - **TOU:**
 - **SQI:**
 - **Set Function:** Sum
 - **Tolerance Value:** 1000
 - **Comparison Operator:** <=
 - **Exception Severity:** Information

Usage Subscriptions and Usage Calculation

This section provides configuration examples for setting up data related to usage subscriptions and calculations, including usage requests from an external system (such as Oracle Utilities Customer Care and Billing) and rate comparison scenarios.

Usage Subscription Example

Scenario: Configure the data needed to support usage calculations for a residential electric service point where a manually-read scalar meter is installed (see **Service Point / Install Event Examples** on page 16-19), as follows:

- Usage calculations for this service point are performed by usage rules in a group called "KWH - Residential" (KWH_RES).

- Usage transactions created from usage subscriptions of this type are sent to "Oracle Utilities Customer Care and Billing"
- In Customer Care and Billing, the usage subscription is identified as "RES-123" .

Object	Data and Field Values
Usage Subscription Type Business Object: D2-UsageSubscriptionType	<ul style="list-style-type: none"> • Usage Subscription Type: E-RES • Description: Electric Residential • Usage Subscription Business Object: Usage Subscription • Usage Recipient: Customer Care and Billing • Subscription Type: CIS • Valid Service Point Types: <ul style="list-style-type: none"> • SP Type: Electric Residential • Valid Usage Groups: <ul style="list-style-type: none"> • Usage Group: KWH - Residential (KWH_RES)
Usage Subscription Business Object: D2-UsageSubscription	<ul style="list-style-type: none"> • Usage Subscription Type: Electric Residential • Status: Active • Start Date/Time: 12-15-2013 12:00:00AM • Usage Recipient: Customer Care and Billing • External ID: RES-123 • Time Zone: US Pacific Time • Usage Groups: <ul style="list-style-type: none"> • Effective Date/Time: 01-01-2014 12:00:00AM • Usage Group: KWH - Residential (KWH_RES) • Fallback Usage Groups: <ul style="list-style-type: none"> • Effective Date: 01-01-2014 • Usage Group: KWH - Residential (KWH_RES) • Service Points: <ul style="list-style-type: none"> • Service Point: 1401 Flower Street, Glendale, CA, 99999, US / Electric Residential / Active • Start Date/Time: 01-01-2014 12:00:00AM • Usage: Add • Use Percent: 100

Usage Subscription Example - Usage Request from External System

Scenario: Configure the data needed to support usage calculations for a residential electric service point where a manually-read scalar meter is installed (see **Service Point / Install Event**

Examples on page 16-19), as follows:

- Usage calculations for this service point are based on the customer's rate as defined in Oracle Utilities Customer Care and Billing. The customer's rate is "E-RES" which corresponds to a usage group called "Electric Residential KWH" (E-RES).
- The usage group to execute is determined via an "override" algorithm based on the rate specified in the usage request
- Usage transactions created from usage subscriptions of this type are sent to "Oracle Utilities Customer Care and Billing"
- In Customer Care and Billing, the usage subscription is identified as "RES-321" .

Object	Data and Field Values
CCB Rate Schedule Extendable Lookup Business Object: D2-CCBRateScheduleLookup	<ul style="list-style-type: none"> • Rate: E-RES • Description: Electric Residential Rate • Default Usage Group: Electric Residential KWH (E-RES)
Usage Subscription Type Business Object: D2-UsageSubscriptionType	<ul style="list-style-type: none"> • Usage Subscription Type: E-RES-CIS • Description: Electric Residential (via CIS) • Usage Subscription Business Object: Usage Subscription • Usage Recipient: Customer Care and Billing • Subscription Type: CIS • Usage Group Determination Override Algorithm: Derive Usage Group Based On Rate and Installed Device • Valid Service Point Types: <ul style="list-style-type: none"> • SP Type: Electric Residential • Valid Usage Recipients: <ul style="list-style-type: none"> • Usage Recipient: Customer Care and Billing • Valid Usage Groups: <ul style="list-style-type: none"> • Usage Group: Electric Residential KWH (E-RES) • Fallback Usage Groups: <ul style="list-style-type: none"> • Effective Date: 01-01-2014 • Usage Group: Electric Residential KWH (E-RES)

Object	Data and Field Values
Usage Subscription Business Object: D2-UsageSubscription	<ul style="list-style-type: none"> • Usage Subscription Type: Electric Residential (via CIS) • Status: Active • Start Date/Time: 12-15-2013 12:00:00AM • Usage Recipient: Customer Care and Billing • External ID: RES-321 • Time Zone: US Pacific Time • Service Points: <ul style="list-style-type: none"> • Service Point: 1401 Flower Street, Glendale, CA, 99999, US / Electric Residential / Active • Start Date/Time: 01-01-2014 12:00:00AM • Usage: Add • Use Percent: 100

Usage Subscription Example - Rate Comparison via Customer Self Service

Scenario: Configure the data needed to support rate comparison usage calculations for a residential electric service point where an interval meter is installed (service point type: "Electric Residential - Interval"), as follows:

- Usage calculations for this service point are based on the customer's rate as defined in Oracle Utilities Customer Care and Billing. The customer's rate is "E-INT-RES" which corresponds to a usage group called "Electric Residential Interval KWH" (E-INT-RES).
- The usage group to execute is determined via an "override" algorithm based on the rate specified in the usage request.
- Usage transactions created from usage subscriptions of this type are sent to "Oracle Utilities Customer Care and Billing"
- In Customer Care and Billing, the usage subscription is identified as "RES-321"
- Rate comparison can be performed based on electric vehicle use, by selecting one of a pre-defined set of electric vehicle types (defined via a characteristic type and related factor/factor values, with profile measurements for each).

Usage Subscription Data

Object	Data and Field Values
CCB Rate Schedule Extendable Lookup Business Object: D2-CCBRateScheduleLookup	<ul style="list-style-type: none"> • Rate: E-INT-RES • Description: Electric Residential Interval Rate • Default Usage Group: Electric Residential Interval KWH (E-INT-RES)

Object	Data and Field Values
Usage Subscription Type Business Object: D2-UsageSubscriptionType	<ul style="list-style-type: none"> • Usage Subscription Type: E-RES-INT-CIS • Description: Electric Residential - Interval (via CIS) • Usage Subscription Business Object: Usage Subscription • Usage Recipient: Customer Care and Billing • Subscription Type: CIS • Usage Group Determination Override Algorithm: Derive Usage Group Based On Rate and Installed Device • Valid Service Point Types: <ul style="list-style-type: none"> • SP Type: Electric Residential - Interval • Valid Usage Recipients: <ul style="list-style-type: none"> • Usage Recipient: Customer Care and Billing • Valid Usage Groups: <ul style="list-style-type: none"> • Usage Group: Electric Residential Interval KWH (E-INT-RES) • Fallback Usage Groups: <ul style="list-style-type: none"> • Effective Date: 01-01-2014 • Usage Group: Electric Residential Interval KWH (E-INT-RES)
Usage Subscription Business Object: D2-UsageSubscription	<ul style="list-style-type: none"> • Usage Subscription Type: Electric Residential - Interval (via CIS) • Status: Active • Start Date/Time: 12-15-2013 12:00:00AM • Usage Recipient: Customer Care and Billing • External ID: RES-321 • Time Zone: US Pacific Time • Service Points: <ul style="list-style-type: none"> • Service Point: 1401 Flower Street, Glendale, CA, 99999, US / Electric Residential - Interval / Active • Start Date/Time: 01-01-2014 12:00:00AM • Usage: Add • Use Percent: 100

Rate Comparison Data

Note: The base package measuring components used in this example require a device configuration, but in practice, a custom measuring component business object could be used that does not require a device and device configuration.

Object	Data and Field Values
Characteristic Type	<ul style="list-style-type: none"> • Characteristic Type: ELEC_VEH • Description: Electric Vehicles • Type of Char Value: Predefined Value • Characteristic Values: <ul style="list-style-type: none"> • Nissan Leaf <ul style="list-style-type: none"> • Characteristic Value: LEAF • Description: Nissan Leaf • Tesla Model S <ul style="list-style-type: none"> • Characteristic Value: TESLA • Description: Tesla Model S • Chevrolet Volt <ul style="list-style-type: none"> • Characteristic Value: VOLT • Description: Chevrolet Volt
Measuring Component Type Business Object: D1-IntervalChannelTypePhysical	<ul style="list-style-type: none"> • Measuring Component Type: KWH-60MIN • Description: KWH - 60 Minutes • Measuring Component Business Object: Interval Channel • Measurement Business Object: Measurement • Service Type: Electric • Consumptive / Subtractive: Consumptive • Seconds Per Interval: 01:00:00
Device Configuration Type Business Object: D1-DeviceConfigurationType	<ul style="list-style-type: none"> • Device Configuration Type: KWH-INTERVAL • Description: KWH - Interval Channel • Device Configuration Business Object: Device Configuration • Service Type: Electric • Valid Measuring Component Types: <ul style="list-style-type: none"> • KWH - 60 Minutes

Object	Data and Field Values
Device Type Business Object: D1-SmartMeterType	<ul style="list-style-type: none"> • Device Type: ELEC-SMART-INT-KWH • Description: Electric Smart Meter - Interval KWH • Device Business Object: Smart Meter • Service Type: Electric • Device Classification: Meter • Valid Device Configuration Types: <ul style="list-style-type: none"> • KWH - Interval Channel
Device Business Object: D1-SmartMeter	<ul style="list-style-type: none"> • Device Type: Electric Smart Meter - Interval KWH • Serial Number: 12345 • Status: Active
Measuring Component (Nissan Leaf) Business Object: D1-IntervalChannel	<ul style="list-style-type: none"> • Measuring Component Type: KWH - 60 Minutes • Device Configuration: Electric Smart Meter - Interval KWH / Effective Date/Time: 01-01-2014 12:00:00AM / KWH - Interval Channel / 3 Measuring Component(s) / Active • Status: Active • How To Use: Additive • External ID: Nissan Leaf
Measuring Component (Tesla Model S) Business Object: D1-IntervalChannel	<ul style="list-style-type: none"> • Measuring Component Type: KWH - 60 Minutes • Device Configuration: Electric Smart Meter - Interval KWH / Effective Date/Time: 01-01-2014 12:00:00AM / KWH - Interval Channel / 3 Measuring Component(s) / Active • Status: Active • How To Use: Additive • External ID: Tesla Model S
Measuring Component (Chevrolet Volt) Business Object: D1-IntervalChannel	<ul style="list-style-type: none"> • Measuring Component Type: KWH - 60 Minutes • Device Configuration: Electric Smart Meter - Interval KWH / Effective Date/Time: 01-01-2014 12:00:00AM / KWH - Interval Channel / 3 Measuring Component(s) / Active • Status: Active • How To Use: Additive • External ID: Chevrolet Volt

Object	Data and Field Values
Factor Business Object: D1-FactorProfile	<ul style="list-style-type: none"> • Factor: ELECTRIC_VEHICLE • Description: Electric Vehicle • Factor Class: Profile • Characteristic Source Algorithm: Factor Characteristic Source N/A Algorithm • Factor Characteristic Type: Electric Vehicle
Factor Value (Nissan Leaf) Business Object: D1-FactorValueProfile	<ul style="list-style-type: none"> • Factor: Electric Vehicle • Factor Characteristic Type: Electric Vehicle • Factor Characteristic Value: LEAF • Effective Date/Time: 01-01-2014 12:00:00AM • Profile: 12345/ Nissan Leaf / KWH - 60 Minutes
Factor Value (Tesla Model S) Business Object: D1-FactorValueProfile	<ul style="list-style-type: none"> • Factor: Electric Vehicle • Factor Characteristic Type: Electric Vehicle • Factor Characteristic Value: LEAF • Effective Date/Time: 01-01-2014 12:00:00AM • Profile: 12345/ Tesla Model S/ KWH - 60 Minutes
Factor Value (Chevrolet Volt) Business Object: D1-FactorValueProfile	<ul style="list-style-type: none"> • Factor: Electric Vehicle • Factor Characteristic Type: Electric Vehicle • Factor Characteristic Value: LEAF • Effective Date/Time: 01-01-2014 12:00:00AM • Profile: 12345/ Chevrolet Volt / KWH - 60 Minutes
Usage Adjustment Type Extendable Lookup Business Object: D2-UsageAdjustmentTypeLookup	<ul style="list-style-type: none"> • Usage Adjustment Type: ELEC_VEHICLE • Description: Purchase of Electric Vehicle • Override Description: Purchase of Electric Vehicle • External Reference ID: Purchase of Electric Vehicle

Object	Data and Field Values
<p>Self-Service Master Configuration Business Object: WX-MDMMasterConfig</p> <p>Note: For this example, only Rate Comparison Configuration details are provided. See Self Service Master Configuration on page 2-6 for more information about this master configuration).</p>	<ul style="list-style-type: none">• Factor Characteristic Type Indicating No Value Variation: N/A• External Reference Factor Value Characteristic Type: External Reference ID• Minimum Days of Usage Adjustment Data: 2• Rate / Usage Adjustments:<ul style="list-style-type: none">• Rate: Electric Residential - Interval KWH• Usage Adjustments:<ul style="list-style-type: none">• Usage Adjustment Factor: Electric Vehicle• Usage Adjustment Type: Purchase of Electric Vehicle

Sample Implementation

This section describes the steps involved in configuring Oracle Utilities Meter Data Management in a simple example implementation, including the following:

- **Implementation Description and Requirements**
- **Implementation Steps**

Note: The implementation described in this section is intended for example purposes only, and is intentionally simple, and as such does not involve configuration of every type of object described in this book. Also, this example assumes that the base package admin business objects (device type, measuring component type, etc. meet the requirements of the implementation.

Implementation Description and Requirements

The sample implementation described in this section will be for a small electric utility providing service to a small town that includes residential, commercial, and industrial customers. The details and requirements of this implementation are summarized as follows

Requirement	Description
Types of Customers	<ul style="list-style-type: none"> • Residential (approximately 50,000) • Commercial (approximately 1,000) • Industrial (approximately 100)
Types of Meters	<ul style="list-style-type: none"> • Residential: Scalar, single register (manual) • Commercial: Interval channel/scalar register (smart) • Industrial: Two interval channels (smart)
Meter Manufacturers / Head-End Systems	<ul style="list-style-type: none"> • MetersRUs: Scalar (used for residential customers) • MeterTech, Inc.: Interval (used for commercial and industrial customer)
Usage Metered	<ul style="list-style-type: none"> • Residential: KWH (scalar) • Commercial: KWH (interval and scalar) • Industrial: KWH and KVARH (interval)
Readings/Measurement Data	<ul style="list-style-type: none"> • Scalar (for residential and commercial) • Interval (for commercial and industrial, 15 minutes)

Requirement	Description
Validation, Editing, and Estimation Rules	<p>All meters:</p> <ul style="list-style-type: none"> • Meter Multiplier Check • UOM Check • Device ID Check <p>Interval meters:</p> <ul style="list-style-type: none"> • Interval spike check • Interval size validation • Interval replacement • Sum Check (commercial only) • KVARH Check (industrial only) - identifies intervals where reactive load (kVARh) is present and active load (kWh) is not. <p>Scalar meters:</p> <ul style="list-style-type: none"> • Scalar Replacement • Negative consumption
Bill Determinants	<ul style="list-style-type: none"> • Residential: KWH • Commercial: KWH and KW (demand) • Industrial: On-Peak KWH, Off-Peak KWH, and KVA (demand)
Billing System	Oracle Utilities Customer Care and Billing

Implementation Steps

This section outlines the steps in configuring Oracle Utilities Meter Data Management to meet the requirements outlined above. These steps include:

1. Design and Create Business Objects

In this first step, we'll outline the specific business objects and other configuration data required to address the sample requirements.

2. Create Admin Data

In this step, we'll outline the admin data that would need to be created to address the sample requirements.

3. Create Master Data

In this step, we'll outline the master data (individual devices, service points, etc.) that would need to be created to address the sample requirements.

Design and Create Business Objects

The first step in implementing and configuring the system is to identify the business objects (and related configuration objects) needed to meet the requirements of the implementation. This section outlines the business objects and other significant configuration objects that could be created to meet the requirements of our sample implementation. This does NOT include listings of all configuration objects needed (such as individual display and maintenance UI maps, portal navigation options, etc.).

Service Points and Device Installation

For service point and device installation data, we would need to create the following:

Service Points: Service point business objects for each type of customer, as follows:

- Residential: CM-ResidentialSP
- Commercial: CM-CommercialSP
- Industrial: CM-IndustrialSP

Contacts: Contact business objects for each type of customer, as follows:

- Residential: CM-ResPerson
- Commercial: CM-ComBusiness
- Industrial: CM-IndBusiness

Install Events: This implementation can use the base package install event business objects, as follows:

- Smart Meters: D1-SmartMeterInstallEvent
- Manual Meters: D1-ManualMeterInstallEvent

Service Providers: This implementation can use the base package service provider business objects, as follows:

- Head-End Systems: D1-HeadEndSystem
- Billing System: D1-ExternalApplication

Activities: This implementation can use the base package activity business objects.

Devices and Measuring Components

For devices and measuring components, we would need to create the following:

Devices: Device business objects for each type of meter, as follows:

- Residential: CM-ScalarRegister
 - Install Event BO: D1-ManualMeterInstallEvent (see above)
- Commercial: CM-IntChanScalarReg
 - Install Event BO: D1-SmartMeterInstallEvent(see above)
- Industrial: CM-Interval2Channels
 - Install Event BO: D1-SmartMeterInstallEvent (see above)

Measuring Components: Measuring component business objects for scalar registers and/or interval channels, as follows:

- Residential - Scalar Register: CM-ResScalarRegister
- Commercial/Industrial - Interval Channel: CM-IntervalChannel (used for both commercial and industrial meters)
- Commercial - Scalar Register: CM-ScalarValRegister

Measurement Data

For measurement data, we would need to create the following:

Initial Measurement Data: Initial load, estimation, and manual initial measurement business objects for each reading/measurement type, as follows:

- Initial Load - Interval: CM-InitialLoadIMDInterval
- Initial Load - Scalar: CM-InitialLoadIMDScalar
- Estimation - Interval: CM-EstimationIMDInterval
- Estimation - Scalar: CM-EstimationIMDScalar
- Manual - Interval: CM-ManualIMDInterval
- Manual - Scalar: CM-ManualIMDScalar

Measurement: A single measurement business object for all final measurements, as follows:

- Final Measurement: CM-FinalMeasurement

VEE Groups and Rules

For VEE groups and rules, we would need to create the following:

VEE Rules: Business object and algorithm type / algorithm for the KVARH Check validation, as follows:

- Business Object: CM-KVARHCheck
- Algorithm Type: CM-KVARHCHK
- Algorithm: CM-KVARHCHK

Usage Subscriptions

For usage subscriptions, we would need to create the following:

Usage Subscriptions: This implementation can use the base package usage subscription business object, as follows:

- D2-UsageSubscription

Usage Groups and Rules

For usage groups and rules, we would need to create the following:

Usage Rules: Business object and algorithm type / algorithm for the KVA calculation, as follows:

- Business Object: CM-CalculateKva
- Algorithm Type: CM-CALCKVA
- Algorithm: CM-CALCKVA

TOU Maps and Dynamic Options

For TOU maps and dynamic options, this implementation can use the base package business objects.

Usage Transactions

For usage transactions, we would need to create the following:

Usage Transactions: This implementation can use the base package usage transaction business objects, as follows:

- D2-UsageTransaction
- D2-SubUsageTransaction

Create Admin Data

With all of the custom business objects needed for the implementation in place, the next step would be to create admin data. This section outlines the admin data that would need to be created to meet the requirements of our sample implementation. In general these listings list only the name (or code) and description of each record to be created, and do not include details for every attribute of each record created. Where listing additional attributes is important to understanding how the data would be created, it is noted, and additional details are provided in a separate section.

The table below summarizes the common admin data needed for our implementation

Admin Data Type	Data to Create
Activity Type	N/A (will use base package activity types)
Contact Type	<p>One for each type of customer rule, as follows:</p> <ul style="list-style-type: none"> • RESIDENTIAL (Residential - Person) Business Object: CM-ResPerson • COMMERCIAL (Commercial - Business) Business Object: CM-ComBusiness • INDUSTRIAL (Industrial - Business) Business Object: CM-IndBusiness

Exception Type	<p>One for each VEE rule, as follows:</p> <ul style="list-style-type: none"> • MMULTCHK (Meter Multiplier Check) • UOMCHK (UOM Check) • DVCICCHK (Device ID Check) • INTSPIKECHK (Interval spike check) • INTSIZEVAL (Interval size validation) • INTREPL (Interval replacement) • INTKVARHCHK (Interval KVARH Check) • SCALREPL (Scalar Replacement) • NEGCONS (Negative consumption)
Factor	N/A
Market	<p>Single market:</p> <ul style="list-style-type: none"> • SMALLTOWNUSA
Measurement Cycle	<p>Twenty cycles for each type of customer, as follows:</p> <ul style="list-style-type: none"> • RESMC01, RESMC02, ..., RESMC20 • COMMC01, COMMC02, ..., COMMC20 • INDMC01, INDMC02, ..., INDMC20
Measurement Cycle Schedule	<p>One per measurement cycle per month, as follows:</p> <ul style="list-style-type: none"> • RESMC01: <ul style="list-style-type: none"> • Scheduled Selection Date: 08/02/2010 • Expected Work Date: 08/03/2010 • RESMC02: <ul style="list-style-type: none"> • Scheduled Selection Date: 08/03/2010 • Expected Work Date: 08/04/2010 • Etc.
Service Provider	<p>One for each head-end system, and one for the billing system, as follows:</p> <ul style="list-style-type: none"> • METERSRUS (Meters R Us) Business Object: D1-HeadEndSystem • METERTECH (MeterTech, Inc.) Business Object: D1-HeadEndSystem • OUCCB (Oracle Utilities Customer Care and Billing) Business Object: D1-ExternalApplication <p>* See Service Providers on page 16-66 for additional details.</p>
Service Quantity Identifier	N/A
Service Type	<p>Single service type:</p> <ul style="list-style-type: none"> • ELECTRIC (Electric)
Time of Use	<p>One for each time of use period, as follows:</p> <ul style="list-style-type: none"> • ONPEAK (On Peak) • OFFPEAK (Off Peak)

VEE Group	See VEE Groups and Rules on page 16-67 for details.
Dynamic Option Type	N/A
Manufacturer	<p>One for each manufacturer, as follows:</p> <ul style="list-style-type: none"> METERSRUS (Meters R Us) <ul style="list-style-type: none"> Models: RES2010 METERTECH (MeterTech, Inc.) <ul style="list-style-type: none"> Models: COM2010, IND2010
TOU Group	<p>One group for TOU periods, as follows:</p> <ul style="list-style-type: none"> ON-OFF_TOU_GRP (On-Peak / Off Peak TOU Group) - Contains ONPEAK and OFFPEAK TOU periods.
Unit of Measure	<p>One for each type of metered usage and for calculated usage, as follows:</p> <ul style="list-style-type: none"> KVA (calculated) KVAR (used in KVA calculation) KVARH (measured) KW (derived) KWH (measured) <p>* All UOMs use the ELECTRIC service type.</p>
VEE Rule	See VEE Groups and Rules on page 16-67 for details.
Measuring Component Type	<p>Five measuring component types, as follows:</p> <ul style="list-style-type: none"> RESSCALAR (Residential Scalar Register) <p>Business Object: CM-ResScalarRegister</p> COMINTERVAL (Commercial Interval Channel) <p>Business Object: CM-IntervalChannel</p> COMSCALAR (Commercial Scalar Register) <p>Business Object: CM-ScalarValRegister</p> INDINTERVALKVARH (Industrial KVARH Interval Channel) <p>Business Object: CM-IntervalChannel</p> INDINTERVALKWH (Industrial KWH Interval Channel) <p>Business Object: CM-IntervalChannel</p> <p>* All measuring component types use the ELECTRIC service type. See Measuring Component Types on page 16-68 for additional details.</p>
TOU Map Template	<p>One TOU map template for each TOU schedule, as follows:</p> <ul style="list-style-type: none"> IND_TOU_SCHED_1 (TOU Schedule 1) IND_TOU_SCHED_2 (TOU Schedule 2) Etc. <p>* See TOU Map Templates and TOU Map Types on page 16-70 for additional details.</p>

Device Configuration Type	<p>One device configuration type for each type of meter installed, as follows:</p> <ul style="list-style-type: none"> • RESDVCCFG (Residential Device Configuration) <p>Valid Measuring Component Types:</p> <ul style="list-style-type: none"> • RESSCALAR (Residential Scalar Register) <ul style="list-style-type: none"> • COMDVCCFG (Commercial Device Configuration) <p>Valid Measuring Component Types:</p> <ul style="list-style-type: none"> • COMSCALAR (Commercial Scalar Register) • COMINTERVAL (Commercial Interval Channel) <ul style="list-style-type: none"> • INDDVCCFG (Industrial Device Configuration) <p>Valid Measuring Component Types:</p> <ul style="list-style-type: none"> • INDINTERVALKVARH (Industrial KVARH Interval Channel) • INDINTERVALKWH (Industrial KWH Interval Channel) <p>*All device configuration types use the ELECTRIC service type.</p>
Device Type	<p>One device type for each type of meter installed, as follows:</p> <ul style="list-style-type: none"> • RESDVC (Residential Device) <p>Business Object: CM-ScalarRegister</p> <p>Valid Device Configuration Types:</p> <ul style="list-style-type: none"> • RESDVCCFG (Residential Device Configurations) <ul style="list-style-type: none"> • COMDVC (Commercial Device) <p>Business Object: CM-IntChanScalarReg</p> <p>Valid Device Configuration Types:</p> <ul style="list-style-type: none"> • COMDVCCFG (Commercial Device Configurations) <ul style="list-style-type: none"> • INDDVC (Industrial Device) <p>Business Object: CM-Interval2Channels</p> <p>Valid Device Configuration Types:</p> <ul style="list-style-type: none"> • INDDVCCFG (Industrial Device Configurations) <p>*All device types use the ELECTRIC service type.</p>
TOU Map Type	<p>One TOU map type for each of three interval sizes, as follows:</p> <ul style="list-style-type: none"> • IND_TOU_MAP_15_MIN (TOU Map Type - 15 Minutes) • IND_TOU_MAP_30_MIN (TOU Map Type - 30 Minutes) • IND_TOU_MAP_60_MIN (TOU Map Type - 60 Minutes) <p>* See TOU Map Templates and TOU Map Types on page 16-70 for additional details.</p>

Usage Group	<p>One usage group for each type of customer, as follows:</p> <ul style="list-style-type: none"> RES_USAGE_GRP (Residential Usage Rules) <p>Valid Device Configuration Types:</p> <ul style="list-style-type: none"> RESDVCCFG (Residential Device Configurations) <ul style="list-style-type: none"> COM_USAGE_GRP (Commercial Usage Rules) <p>Valid Device Configuration Types:</p> <ul style="list-style-type: none"> COMDVCCFG (Commercial Device Configurations) <ul style="list-style-type: none"> IND_USAGE_GRP (Residential Usage Rules) <p>Valid Device Configuration Types:</p> <ul style="list-style-type: none"> INDDVCCFG (Industrial Device Configurations) <p>* See Usage Groups and Rules on page 16-71 for additional details.</p>
Service Point Type	<p>One for each type of customer rule, as follows:</p> <ul style="list-style-type: none"> RESIDENTIAL (Residential) <p>Business Object: CM-ResidentialSP</p> <p>Valid Device Types:</p> <ul style="list-style-type: none"> RESDVC (Residential Devices) <ul style="list-style-type: none"> COMMERCIAL (Commercial) <p>Business Object: CM-CommercialSP</p> <p>Valid Device Types:</p> <ul style="list-style-type: none"> COMDVC (Commercial Devices) <ul style="list-style-type: none"> INDUSTRIAL (Industrial) <p>Business Object: CM-IndustrialSP</p> <p>Valid Device Types:</p> <ul style="list-style-type: none"> INDDVC (Industrial Devices) <p>* All service point types use ELECTRIC service type</p>
Usage Rule	<p>Five usage rules, as follows:</p> <ul style="list-style-type: none"> RES_SCALAR_DETAILS (Retrieve KWH for Scalar Register) COM_APPLY_MATH_KWH (Calculate KWH for Interval Channel) COM_APPLY_MATH_KW (Calculate KW for Interval Channel) IND_GET_TOU_DATA (Calculate TOU-based KWH for interval channel) CALC_KVA (Calculate KVA from KWH and KVARH interval channels) <p>Business Object: CM-CalculateKva</p> <p>* See Usage Groups and Rules on page 16-71 for additional details.</p>

Usage Subscription Type	<p>One for each type of customer rule, as follows:</p> <ul style="list-style-type: none"> RESIDENTIAL (Residential) <p>Business Object: D2-UsageSubscription</p> <p>Usage Recipient: OUCCB (Oracle Utilities Customer Care and Billing)</p> <ul style="list-style-type: none"> COMMERCIAL (Commercial) <p>Business Object: D2-UsageSubscription</p> <p>Usage Recipient: OUCCB (Oracle Utilities Customer Care and Billing)</p> <ul style="list-style-type: none"> INDUSTRIAL (Industrial) <p>Business Object: D2-UsageSubscription</p> <p>Usage Recipient: OUCCB (Oracle Utilities Customer Care and Billing)</p>
-------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

* See **Usage Subscription Types** on page 16-72 for additional details.

Additional Details

This section provides additional details related to the admin data described above. Not all attributes are listed for all types of data.

Service Providers

This section provides additional details for each of the service providers listed above.

Service Provider: METERSRUS

- Business Object: D1-HeadEndSystem
- Description: Meters R Us
- External Reference ID: HE-MRU
- Our Name/ID in Their System: HE-MRU-11
- Processing Methods List:
 - Initial Measurement Creation (How To Create MC Related Information)
 - Default Processing Method (Business Object): CM-InitialLoadIMDScalar
 - Override Process Method:

Measuring Component Type	Business Object
Residential Scalar Register (RESSCALAR)	CM-InitialLoadIMDScalar

Service Provider: METERTECH

- Business Object: D1-HeadEndSystem
- Description: MeterTech, Inc.
- External Reference ID: HE-MTECH
- Our Name/ID in Their System: HE-MTECH-11
- Processing Methods List:
 - Initial Measurement Creation (How To Create MC Related Information)
 - Default Processing Method (Business Object): CM-InitialLoadIMDInterval

- Override Process Method:

Measuring Component Type	Business Object
Commercial Interval Channel (COMINTERVAL)	CM-InitialLoadIMDInterval
Commercial Scalar Register (COMSCALAR)	CM-InitialLoadIMDScalar
Industrial KVARH Interval Channel (INDINTERVALKVARH)	CM-InitialLoadIMDInterval
Industrial KWH Interval Channel (INDINTERVALKWH)	CM-InitialLoadIMDInterval

Service Provider: OUCCB

- Business Object: D1-ExternalApplication
- Description: Oracle Utilities Customer Care and Billing
- External Reference ID: EXT-CCB
- Our Name/ID in Their System: EXT-CCB-11
- Processing Methods List:
 - Usage Transaction Creation (How To Create Usage Subscription Related Information)
 - Default Processing Method (Business Object): D2-UsgTranSeeder

VEE Groups and Rules

The table below lists the VEE groups and corresponding VEE rules for this implementation.

VEE Group	VEE Rules
ALL_RULES Contains rules for all measuring components (referenced by VEE rule).	<ul style="list-style-type: none"> • MMULTCHK (Meter Multiplier Check) • UOMCHK (UOM Check) • DVCICCHK (Device ID Check)
INTERVAL_RULES Contains rules for interval measuring components (referenced by VEE rule).	<ul style="list-style-type: none"> • INTSPIKECHK (Interval Spike) • INTSIZEVAL (Interval Size Val) • INTREPL (Interval Replacement)
SCALAR_RULES Contains rules for scalar measuring components (referenced by VEE rule).	<ul style="list-style-type: none"> • SCALREPL (Scalar Replacement) • NEGCONS (Negative Consumption)
SCALAR_MCS Contain rules to be applied to scalar measuring components (including referenced groups).	<ul style="list-style-type: none"> • ALL_REF (References ALL_RULES) • SCALAR_REF (References SCALAR_RULES) • SCALAR_EXCEPTIONS (Exception Handler for Scalar)

VEE Group	VEE Rules
COM_INTD_MCS Contain rules to be applied to commercial interval measuring components (including referenced groups).	<ul style="list-style-type: none"> • ALL_REF (References ALL_RULES) • INTERVAL_REF (References INTERVAL_RULES) • SUMCHK (Sum Check) • INTD_EXCEPTIONS (Exception Handler for Interval)
IND_INTD_MCS Contain rules to be applied to industrial interval measuring components (including referenced groups).	<ul style="list-style-type: none"> • ALL_REF (References ALL_RULES) • INTERVAL_REF (References INTERVAL_RULES) • INTKVARHCHK (Interval KVARH Check) • INTD_EXCEPTIONS (Exception Handler for Interval)

Measuring Component Types

This section provides additional details for each of the measuring component types listed above.

Measuring Component Type: RESSCALAR

- Description: Residential Scalar Register
- Measuring Component Business Object: CM-ResScalarRegister
- Measurement Business Object: CM-FinalMeasurement
- Service Type: Electric
- Value Identifiers:

Value Identifier Type	Short-Hand Description	UOM
Measurement	KWH	KWH

- Valid VEE Groups:
 - Initial Load VEE Group - Scalar (SCALAR_MCS)
- Fallback VEE Groups:
 - Initial Load: Initial Load VEE Group - Scalar (SCALAR_MCS)

Measuring Component Type: COMINTERVAL

- Description: Commercial Interval Channel
- Measuring Component Business Object: CM-IntervalChannel
- Measurement Business Object: CM-FinalMeasurement
- Service Type: Electric

- Value Identifiers:

Value Identifier Type	Short-Hand Description	UOM
Measurement	KWH	KWH

- Valid VEE Groups:
 - Initial Load VEE Group - Commercial Interval (COM_INTD_MCS)
- Fallback VEE Groups:
 - Initial Load: Initial Load VEE Group - Commercial Interval (COM_INTD_MCS)

Measuring Component Type: COMSCALAR

- Description: Commercial Scalar Register
- Measuring Component Business Object: CM-ScalarValRegister
- Measurement Business Object: CM-FinalMeasurement
- Service Type: Electric
- Value Identifiers:

Value Identifier Type	Short-Hand Description	UOM
Measurement	KWH	KWH

- Valid VEE Groups:
 - Initial Load VEE Group - Scalar (SCALAR_MCS)
- Fallback VEE Groups:
 - Initial Load: Initial Load VEE Group - Scalar (SCALAR_MCS)

Measuring Component Type: INDINTERVALKVARH

- Description: Industrial KVARH Interval Channel
- Business Object: CM-IntervalChannel
- Measurement Business Object: CM-FinalMeasurement
- Service Type: Electric
- Value Identifiers:

Value Identifier Type	Short-Hand Description	UOM
Measurement	KVARH	KVARH

- Valid VEE Groups:
 - Initial Load VEE Group - Industrial Interval (IND_INTD_MCS)
- Fallback VEE Groups:
 - Initial Load: Initial Load VEE Group - Industrial Interval (IND_INTD_MCS)

Measuring Component Type: INDINTERVALKWH (Industrial KWH Interval Channel)

- Description: Industrial KWH Interval Channel
- Business Object: CM-IntervalChannel
- Measurement Business Object: CM-FinalMeasurement
- Service Type: Electric
- Value Identifiers:

Value Identifier Type	Short-Hand Description	UOM
Measurement	KWH	KWH

- Valid VEE Groups:
 - Initial Load VEE Group - Industrial Interval (IND_INTD_MCS)
- Fallback VEE Groups:
 - Initial Load: Initial Load VEE Group - Industrial Interval (IND_INTD_MCS)

TOU Map Templates and TOU Map Types

This section provides additional details for the TOU map templates and TOU map types listed above. In the case where different types of industrial customers use different rules for On Peak vs. Off Peak hours, different TOU map templates could be created for different TOU schedules. In this case, the primary difference between TOU map templates would be in the TOU Schedule Section.

TOU Map Template: IND_TOU_SCHED_1

- Description: TOU Schedule 1
- TOU Group: On-Peak / Off Peak TOU Group (ON-OFF_TOU_GRP)
- Default TOU: Off Peak (OFFPEAK)
- Work Calendar: Small Town USA Work Calendar (SMTUWRKCAL)
- Holiday TOU: Off Peak (OFFPEAK)
- Holiday Template: N/A
- Interval Size: 01:00:00
- TOU Schedule Section: Etc.

For each TOU map template, you might create multiple TOU map types for different interval sizes. For example:

TOU Map Type: IND_TOU_MAP_15_MIN

- Description: TOU Map Type - 15 Minutes (for Usage Calculation)
- TOU Map Business Object: D2-TOUMap
- Time Zone: US - Eastern Time
- Interval Size: 00:15:00
- Default TOU Map Template: TOU Schedule 1 (IND_TOU_SCHED_1)
- Override TOU Map Templates: N/A

TOU Map Type: IND_TOU_MAP_30_MIN

- Description: TOU Map Type - 30 Minutes (for Aggregation)
- TOU Map Business Object: D2-TOUMap
- Time Zone: US - Eastern Time
- Interval Size: 00:30:00
- Default TOU Map Template: TOU Schedule 1 (IND_TOU_SCHED_1)
- Override TOU Map Templates: N/A

TOU Map Type: IND_TOU_MAP_60_MIN

- Description: TOU Map Type - 60 Minutes (for Aggregation)
- TOU Map Business Object: D2-TOUMap
- Time Zone: US - Eastern Time
- Interval Size: 01:00:00
- Default TOU Map Template: TOU Schedule 1 (IND_TOU_SCHED_1)
- Override TOU Map Templates: N/A

Usage Groups and Rules

The table below lists the usage groups and corresponding usage rules for this implementation.

Usage Group	Usage Rules
RES_USAGE_GRP (Residential Usage Rules) Valid Device Configuration Types: <ul style="list-style-type: none"> • RESDVCCFG (Residential Device Configurations) 	<ul style="list-style-type: none"> • RES_SCALAR_DETAILS (Retrieve KWH for Scalar Register)
COM_USAGE_GRP (Commercial Usage Rules) Valid Device Configuration Types: <ul style="list-style-type: none"> • COMDVCCFG (Commercial Device Configurations) 	<ul style="list-style-type: none"> • COM_APPLY_MATH_KWH (Calculate KWH for Interval Channel) • COM_APPLY_MATH_KW (Calculate KW for Interval Channel)
IND_USAGE_GRP (Residential Usage Rules) Valid Device Configuration Types: <ul style="list-style-type: none"> • INDDVCCFG (Industrial Device Configurations) 	<ul style="list-style-type: none"> • IND_GET_TOU_DATA (Calculate TOU-based KWH for interval channel) • CALC_KVA (Calculate KVA from KWH and KVARH interval channels)

Usage Subscription Types

This section provides additional details for each of the subscription types listed above.

Usage Subscription Type: RESIDENTIAL

- Description: Residential
- Usage Subscription Business Object: D2-UsageSubscription
- Usage Recipient: OUCCB (Oracle Utilities Customer Care and Billing)
- Valid Service Point Types:
 - Residential (RESIDENTIAL)
- Valid Usage Recipients:
 - OUCCB (Oracle Utilities Customer Care and Billing)
- Valid Usage Groups:
 - Residential Usage Rules (RES_USAGE_GRP)
- Fallback Usage Groups:

Effective Date	Usage Group
08-01-2010	Residential Usage Rules (RES_USAGE_GRP)

Usage Subscription Type: COMMERCIAL

- Description: Commercial
- Usage Subscription Business Object: D2-UsageSubscription
- Usage Recipient: OUCCB (Oracle Utilities Customer Care and Billing)
- Valid Service Point Types:
 - Commercial (COMMERCIAL)
- Valid Usage Recipients:
 - OUCCB (Oracle Utilities Customer Care and Billing)
- Valid Usage Groups:
 - Commercial Usage Rules (COM_USAGE_GRP)
- Fallback Usage Groups:

Effective Date	Usage Group
08-01-2010	Commercial Usage Rules (COM_USAGE_GRP)

Usage Subscription Type: INDUSTRIAL

- Description: Industrial
- Usage Subscription Business Object: D2-UsageSubscription
- Usage Recipient: OUCCB (Oracle Utilities Customer Care and Billing)
- Valid Service Point Types:

- Industrial (INDUSTRIAL)
- Valid Usage Recipients:
 - OUCCB (Oracle Utilities Customer Care and Billing)
- Valid Usage Groups:
 - Industrial Usage Rules (IND_USAGE_GRP)
- Fallback Usage Groups:

Effective Date	Usage Group
08-01-2010	Industrial Usage Rules (IND_USAGE_GRP)

Create Master Data

In this last step, we would create the actual master data (individual devices, service points, etc.) for the implementation. For purposes of this section, only a single example of each type of data is presented.

Contacts

A typical residential contact might look like this:

Residential - Person:

- Information: John Smith / 555-555-5555
- Contact Type: Residential - Person (RESIDENTIAL)
- Name: John Smith
- Home Phone: 555-555-5555

Service Points

A typical commercial service point might look like this:

Commercial Service Point:

- Information: 35 York Street, Burlington, MA, 01803, US / Commercial / Active
- Service Point Type: Commercial (COMMERCIAL)
- Status: Active
- Time Zone: US - Eastern Time
- Market: Small Town USA
- Main Contact: Phillip Jones
- Address:
 - Country: United States
 - Postal Code: 01803
 - Street Address: 35 York Street
 - City: Burlington
 - State: MA
- Measurement Cycle:
 - Measurement Cycle: Commercial Cycle 01 (COMMC01)
 - Route: Route 2
 - Sequence: 10

Devices

A typical industrial device might look like this:

Industrial Device:

- Information: 123456 / Industrial Device / Install Date/Time: 08-01-2010 12:00 AM / Pending / MeterTech, Inc. / Active
- Device Type: Industrial Device (INDDVC)
- Serial Number: 123456
- Internal Meter Number: 654321
- Pallet Number: 123456
- Manufacturer: MeterTech, Inc.
- Model: IND2010
- Incoming Data Shift: Shifted
- Arming Required: Arming Required
- Head-End System: MeterTech, Inc. (METERTECH)
- Status: Active

Device Configurations

A typical industrial device configuration might look like this:

Industrial Device Configuration:

- Information: Industrial Device / Effective Date/Time: 08-01-2010 12:00 AM / Industrial Device Configuration / 2 Measuring Component(s) / Active
- Device Configuration Type: Industrial Device Configuration (INDDVCCFG)
- Device: 123456 / Industrial Device / Install Date/Time: 08-01-2010 12:00 AM / Pending / MeterTech, Inc. / Active
- Effective Date/Time: 08-01-2010 12:00 AM
- Time Zone: US - Eastern Time
- Status: Active

Measuring Components

Typical industrial measuring components might look like this:

KVARH Interval Channel:

- Information: 123456 / 2 / Industrial KVARH Interval Channel
- Measuring Component Type: Industrial KVARH Interval Channel (INDINTERVALKVARH)
- Device Configuration: Industrial Device / Effective Date/Time: 08-01-2010 12:00 AM / Industrial Device Configuration / 2 Measuring Component(s) / Active
- Consumption Reference Measuring Component: N/A
- How to Use: Consumptive
- Number of Digits Left: 5
- Number of Digits Right: 2
- Channel Multiplier: 1.000
- Latest Read Date/Time: N/A

- Channel ID: 2

KWH Interval Channel:

- Information: 123456 / 1 / Industrial KWH Interval Channel
- Measuring Component Type: Industrial KWH Interval Channel (INDINTERVALKWH)
- Device Configuration: Industrial Device / Effective Date/Time: 08-01-2010 12:00 AM / Industrial Device Configuration / 2 Measuring Component(s) / Active
- Consumption Reference Measuring Component: N/A
- How to Use: Consumptive
- Number of Digits Left: 5
- Number of Digits Right: 2
- Channel Multiplier: 1.000
- Latest Read Date/Time: N/A
- Channel ID: 1

Install Events

A typical industrial installation event might look like this:

Industrial Install Event:

- Information: Install Date/Time: 08-01-2010 / On
- Device Configuration: Industrial Device / Effective Date/Time: 08-01-2010 12:00 AM / Industrial Device Configuration / 2 Measuring Component(s) / Active
- Service Point: 47 North Street, Burlington, MA, 01803, US / Industrial / Active
- Status: On
- Installation Date/Time: 08-01-2010
- Installation Constant: 1.00000
- Device On/Off Status: On
- On/Off History: N/A

TOU Maps

A typical industrial TOU map might look like this:

- Information: Industrial TOU Map
- TOU Map Type: TOU Map Type - 15 Minutes (IND_TOU_MAP_15_MIN)
- Status: Active
- Status Reason: N/A
- Override TOU Map Template: N/A
- Dynamic Option / Dynamic TOU Map Section: N/A
- TOU Data List:
 - 08-01-2010 00:15 AM - Off Peak
 - 08-01-2010 00:30 AM - Off Peak
 - 08-01-2010 00:45 AM - Off Peak
 - 08-01-2010 01:00 AM - Off Peak
 - Etc.

Usage Subscriptions

A typical commercial usage subscription might look like this:

Commercial Usage Subscription:

- Information: Commercial / 08-01-2010 12:00 AM / Active
- Usage Subscription Type: Commercial (COMMERCIAL)
- Status: Active
- Start Date/Time: 08-01-2010 12:00 AM
- End Date/Time: 08-01-2020 12:00 AM
- Usage Recipient: Oracle Utilities Customer Care and Billing (OUCCB)
- Usage Approval: Not Required
- External ID:
- Main Contact: Phillip Jones / Business Phone: 555-555-5555
- Time Zone: US - Eastern Time
- Factor Overrides: N/A
- Usage Groups:

Effective Date/Time	Expiration Date/Time	Usage Group
08-01-2010 12:00 AM	08-01-2011 12:00 AM	Commercial Usage Rules (COM_USAGE_GRP)

- Service Points:

Service Point	Start Date/Time	Stop Date/Time	Usage	Use Percent
35 York Street, Burlington, MA, 01803, US / Commercial / Active	08-01-2010 12:00 AM	08-01-2011 12:00 AM	Add	100

Appendix A

Measurement Services

This appendix provides brief descriptions of the base package measurement services used by VEE rules and measurement functions. The measurement services described in this appendix are implemented as business services. These business services can be used by custom algorithms or BPA/Service scripts created for your implementation.

The table below lists the available back package measurement services, including the business service that implements each service and a brief description for each.

Measurement Service	Business Service	Description
Add Scalar Value To Intervals	D1-AddScalarValueToIntervals	Uses the Apply Formula measurement service to add a scalar value to the value of a specified set of interval data.
Adjust Intervals to Supplied Value	D1-AdjustIntervalsToSuppldVal	Uses the Apply Formula measurement service to adjust the total value of a specified set of interval data to a scalar value.
Apply Formula	D1-ApplyFormula	Used to apply a formula to a specified set of interval data, either by applying a summary function against all intervals of the set, or by manipulating each individual interval in series via a formula using declared constants, or within the context of other sets of input interval data.
Apply TOU Map To Interval Measuring Component	D1-ApplyTOUMapToIntervalMC	Used to apply a TOU map to a set of intervals for a specified measuring component and date/time range, thereby isolating and summarizing those intervals that occurred during a specific time of use.
Axis Conversion	D1-AxisConversion	Used to convert interval data between units of measure (UOMs) and interval sizes (SPIs), including the conversion between peak and consumption-oriented UOMs.
Convert Scalar Consumption To Interval	D1-IntervalizeScalarConsumptn	Used to convert a scalar consumption value to a set of interval measurements.

Measurement Service	Business Service	Description
Create Intervals	D1-CreateIntervals	Used to create interval data based on supplied parameters (UOM, interval size, number of intervals, value, etc.)
Divide Intervals By Scalar Value	D1-DivideIntervalsByScalarVal	Uses the Apply Formula measurement service to divide the values of a specified set of interval data by a scalar value.
Extract Subset of Intervals	D1-ExtractSubsetOfIntervals	Used to extract a subset of interval data from a specified set of intervals.
Identify Spikes	D1-IdentifySpikes	Used to identify spikes in a specified set of interval data based on a spike percentage tolerance.
Insert Intervals	D1-InsertIntervals	Used to insert one or more intervals into a set of interval measurements.
Merge Intervals	D1-MergeIntervals	Used to merge a subset of interval data with a specified set of intervals (where overlaps occur, the subset intervals replace the original intervals).
Multiply Intervals By Scalar Value	D1-MultiplyIntervalsByScalarVal	Uses the Apply Formula measurement service to multiply the values of a specified set of interval data by a scalar value.
Remove Intervals	D1-RemoveIntervals	Used to remove one or more intervals from a set of interval measurements.
Retrieve Interval Consumption	D1-IntvConsumptionRetriever	Used to retrieve one or more interval measurements.
Retrieve Scalar Consumption	D1-ScalarConsumptionRetriever	Used to retrieve one or more scalar measurements.
Set Condition	D1-SetCondition	Used to set the condition (status) code of a specified set of interval data.
Shift Intervals	D1-ShiftIntervals	Used to shift one or more intervals forward or backward in time.
Subtract Scalar Value From Intervals	D1-SubtractScalarValToIntervls	Uses the Apply Formula measurement service to subtract a scalar value from the value of a specified set of interval data.

Use the Business Service portal to view more details concerning these measurement services.

Appendix B

Glossary

This glossary provides definitions of commonly used terms.

360 Degree View - Audit View

A zone that allows users to view an interval measurement curve for a given period overlaid with the count of audit records for each individual measurement. It also allows users to magnify a portion of the curve and see how the measurements looked at different points in time.

360 Degree View - Final Values Overlay

A zone that graphs final measurements for a measuring component, and provides the ability to overlay the graphed data with final measurements from other measuring components. The zone also permits overlaying data from the same measuring component for different time periods, as well as data from measuring components measuring different quantities, such as temperature.

360 Degree View - Time of Use by Day

A zone that displays daily TOU-mapped usage data for a measuring component based on a user-defined time period and TOU Map.

360 Degree View - Time of Use Overlay

A zone that displays an overlay of the TOU periods on a final measurement along with totalized TOU consumption based on a user-defined time period and TOU map.

360 Degree View - Timeline Zone

A zone that displays a timeline of activities and other events for a given service point, device, etc.

Activity Type

Defines properties common to a specific type of activity.

Advanced Metering Infrastructure (AMI)

Refers to systems that measure, collect and analyze energy usage, and interact with advanced devices such as electricity meters, gas meters, heat meters, and water meters, through various communication media either on request (on-demand) or on pre-defined schedules.

Measurement Service - Add Scalar Value To Intervals

Measurement service that uses the Apply Formula measurement service to add a scalar value to the value of a specified set of interval data.

Adjust Intervals To Scalar Quantity

An initial measurement data function used to adjust all interval values within a user-defined time period in an interval measurement such that the total of all interval values equals a user-defined scalar quantity. This function results in a measurement that retains the shape of the original measurement, but has been scaled up or down.

Adjust Intervals To Supplied Value

Measurement service that uses the Apply Formula measurement service to adjust the total value of a specified set of interval data to a scalar value.

Adjust Intervals Using Math

A measuring component consumption function used to adjust all interval values within a user-defined time period using math operations (add, subtract, multiply, or divide).

Aggregation

Measurements that represent an summarization of other measurements from a potentially diverse set of devices. For example, an aggregation may derive the sum of the electric energy of all residential customers in a particular postal code within the utility's service territory.

Aggregator

A class of measuring component that stores measurements that represent an summarization of other measurements from a potentially diverse set of devices. For example, an aggregator may derive the sum of the natural gas consumption of all residential customers in a particular postal code within the utility's service territory.

Apply Formula

Measurement service used to apply a formula to a specified set of interval data, either by applying a summary function against all intervals of the set, or by manipulating each individual interval in series via a formula using declared constants, or within the context of other sets of input interval data.

Apply TOU Map To Interval Measuring Component

Measurement service used to apply a TOU map to a set of intervals for a specified date/time range, thereby isolating and summarizing those intervals that occurred during a specific time of use.

Automatic Meter Reading (AMR)

The technology of automatically collecting consumption, diagnostic, and status data from water meter or energy metering devices (water, gas, electric) and transferring that data to a central database for billing, troubleshooting, and analyzing.

Axis Conversion

Measurement service used to convert interval data between units of measure (UOMs) and interval sizes (SPIs), including the conversion between peak and consumption-oriented UOMs.

Bill Determinants

Measurement data summarized for use by a billing application. Bill determinants can take the form of TOU-mapped interval consumption, scalar consumption, scalar readings, and/or interval consumption obtained via measurements. A common variety of bill determinant is TOU-mapped interval consumption, which reduces a full month's worth of interval data into several buckets of consumption based on time of use.

Command

A communication sent to a device to perform some action on the device, such as Connect, Disconnect, Commission, Decommission, On-Demand Read, or Device Status Check (Ping)

Communication

A record of a message sent between Oracle Utilities Smart Grid Gateway and an external system, such as a head-end system or edge application. Communications can flow both inbound and outbound, and can be both one-way and two-way.

Communication Component Device

Devices that are attached to other devices and provide two-way communication with a head-end system and can send readings to head-end systems and/or other data collection systems.

Communication components are used in situations in which the underlying meter is not capable or not enabled to handle this data. Devices of this sort are sometimes referred to as ERT (Electronic Receiver/Transmitter) meters, or communication modules (for example, the term "gas module" may refer to a communication module attached to a gas meter).

Completion Event

Records used to create or update transactions that reflect the effect of an activity. For example, issuing a commission device command could result in the creation or update of an install event while a read device command could result in the creation of initial measurement data.

Consumption

A measurement by a given device of the amount of energy, water, gas, etc. consumed over a given time period. Synonymous with the term "measurement".

Consumptive

Describes a measuring component for which readings are equivalent to the consumption. For example, if we receive a reading of 400 on January 15 and a reading of 600 on February 15, a consumptive measuring component's consumption between January 15 and February 15 would be 600 (not 200).

Contact

An individual or a business entity with which a company has contact. Each contact must reference a contact type.

Contact - Email

Email addresses related to a contact

Contact - Identifier

Identifiers related to a contact, such as social security number, driver's license number, or the contact's ID in a prior system.

Contact - Name

Names related to a contact

Contact - Phone

Phone numbers related to a contact

Contact Type

Defines the properties of a class of entities (businesses, persons).

Convert Scalar to Interval Consumption

A measuring component consumption function used to convert scalar consumption into interval consumption. The converted consumption is held in a new scratchpad measuring component. The "shape" of the new interval measurement can be based on either a profile or can have a flat distribution.

Convert Scalar Consumption To Interval

Measurement service used to convert a scalar consumption value to a set of interval measurements.

Create Intervals

Measurement service used to create interval data based on supplied parameters (UOM, interval size, number of intervals, value, etc.)

Create Scratchpad Interval Consumption

A measuring component consumption function used save all or some of the final measurements shown in the zone as a new initial measurement for a scratchpad measuring component.

Create/Override

A measuring component consumption function used to create new initial measurement data for a selected measuring component for all or part of a selected time period. This function can either copy existing final measurements (for example, from a profile) or create new measurement data for the IMD it creates.

Demand

The rate at which a commodity is delivered at a given instant or averaged over a designated time. For electricity, demand is often expressed in kilowatts (kW) or kilovolt-amperes (kVa).

Device

A physical or virtual object that holds one or more measuring components that can produce data to be handled by the system. Devices can include meters, substations, transformers, demand response devices, weather stations, etc.

Device Configuration

A specific configuration of a device. Over time, a device can have many configurations. Use of effective-dated device configuration allows the device to retain its identifier(s) even while the quantities it is measuring are changing.

Device Configuration Type

Defines the properties of device configurations of this type, including the valid types of measuring components that can be configured for the device.

Device Event

An event of some sort that has taken place relative to a device. Device events can include power outages, power restorations, tampering alerts, command completion, and other information

Device Status Check

A communication sent to a device to test whether the device is communicating with the network, determine the connection status of the meter, and when possible if there are any known malfunctions

Device Type

Information about a class of devices, including properties that apply to all devices of a type, but can be overridden for an individual device.

Distribution Company (DISCO)

A utility company that constructs and maintains the distribution network that delivers a commodity to customers. Depending upon the regulations within the territory, a distribution company may or may not be responsible for billing the customer.

Divide Intervals By Scalar Value

Measurement service that uses the Apply Formula measurement service to divide the values of a specified set of interval data by a scalar value.

Dynamic Option

Used to specify terms that override how usage is normally calculated - such as a critical peak period that affects the TOU mapping of interval consumption.

Dynamic Option Event

The period of time during which a dynamic option is applicable. A dynamic option may have many events over time.

Dynamic Option Type

Used to define information common to dynamic options of a specific type.

Electronic Receiver/Transmitter (ERT)

Devices that are attached to other devices and provide two-way communication with a head-end system and can send readings to head-end systems and/or other data collection systems.

Exception Type

Defines properties common to many exceptions, including the category of the exception.

Extract Subset of Intervals

Measurement service used to extract a subset of interval data from a specified set of intervals.

Factor

A centrally stored set of values for use in validation rules, bill determinants calculations, and other processes. A factor can have different values depending upon some definable attribute of a system object, such as customer size associated with a service point. The values are effective-dated so that changes over time are retained. Examples of factors can include minimum/ maximum thresholds, loss factors, etc. Classes of factors are defined that can have numeric values (as in the above examples), or values pointing to profile measuring components or VEE groups.

Factor Value

An effective-dated value - either a number, a profile measuring component, a VEE group, or some custom-defined value - assigned to a factor and associated to the value of some attribute of a system object. For example, let's assume that a service point can be classified as residential, commercial, or industrial. The tolerance percentage by which a customer's consumption can exceed last month's consumption can be tighter as the customer's SP increases in size. An example configuration of factor values for a single factor called "tolerance percentage" could be:
Residential - 20% Commercial - 10% Industrial - 5%

Final Measurement

Measurement data that has been validated, and if necessary, edited & estimated, and is ready for use in down-stream processing such as bill determinants calculations. Only one final measurement can exist for a given date/time for a given measuring component; one final measurement exists per interval, and likewise one final measurement exists for each scalar reading. In both cases, the final measurement value stored represents the amount consumed between its date/time and the prior final measurement's date/time.

Function

An online-initiated action applied to measurement data, comprising one or more measurement services.

Head-End System

A system that collects measurement data and meter events for eventual submission to the application. Many devices can communicate to the application through a single head-end system. A utility may have numerous head-end systems through which they communicate with devices.

Identifiers

Names, numbers, or other values used to identify an entity within the system, including devices, measuring components, service points, etc.

Identify Spikes

Measurement service used to identify spikes in a specified set of interval data based on a spike percentage tolerance.

Inbound Communication

Communication sent to Oracle Utilities Smart Grid Gateway from an external system, such as a head-end system or edge application

Initial Measurement Data Function (IMDF)

An online means of manipulating initial measurement data. IMD Functions typically move intervals (shift, insert, remove, etc.) or manipulate their values or conditions.

Insert Intervals

An initial measurement data function used to insert intervals into initial measurement data. Intervals can be inserted at either the beginning or end of the measurement. The end date/time of the measurement is shifted to account for the inserted intervals.

Insert Intervals (service)

Measurement service used to insert one or more intervals into a set of interval measurements.

Inbound Communication

Communication sent to MDF (Service and Measurement Data Foundation) from a head-end system or other external system. Each inbound communication has an associated communication type that defines common properties of the communication.

Independent System Operator (ISO)

The entity charged with reliable operation of the grid and provision of open transmission access to all market participants on a non-discriminatory basis.

Initial Measurement Data (IMD)

A set of one or more readings or measurements that have been loaded into the application, usually in a format that is standard for MDF (Service and Measurement Data Foundation). Over its lifecycle (as pertains to MDM - Meter Data Management), any readings within the IMD are converted into consumption, which is then typically subject to VEE processing and then finalized - meaning stored as final measurements. Only initial measurements can be edited directly by end users of MDM. An IMD for a scalar measuring component will have a single measurement (along with a reading from which the measurement value is derived), while an IMD for an interval measuring component will usually contain multiple interval measurements.

Installation Constant

An installation constant is set to a value other than 1 as an indication that when calculating consumption, the installation requires that measurement data be multiplied by this value to get accurate results.

Installation Event

A device's installation information at a service point. The install event represents both the installation and removal of a device. It also records turning a device on or off while it is installed at a service point.

Installation On and Off History

A single installation event records each time the device is turned on and turned off while it is installed at a service point.

Interval Channel (Measuring Component)

A business object (BO) that represents channels associated to a device.

Interval Channel Type - Physical (Measuring Component Type)

A business object (BO) that maps properties of interval measuring component types for those Measuring Components that are part of physical devices.

Interval Channel Type - Scratchpad (Measuring Component Type)

A business object (BO) that maps properties relevant to stand-alone measuring components functioning as scratchpads for interval data manipulation.

Interval Data

Time-series data in which measurements are captured in pre-defined intervals (5 minutes, 15 minutes, 1 hour, etc.). A set of interval measurements for an interval measuring component composes an individual initial measurement data record.

Interval Data Services

Services used to access and manipulate interval measurements.

Interval Data Snapshot

A copy of the interval data associated with a usage transaction that can be sent to external systems for using in billing calculations (such as real-time pricing). Snapshots are included in outbound messages sent to external systems.

Interval Scratchpad (Measuring Component)

A stand-alone measuring component that provides the user with a means to manipulate measurement data without affecting existing measurements.

Interval Size

The "size" of an interval, representing the length of time between intervals. Interval size is typically measured in seconds-per-interval (SPI).

Manual Meter

A business object (BO) used to model a meter that does not accommodate two-way communications and must be read manually.

Manual Meter Installation Event

A business object (BO) that defines the lifecycle of the installation of a manual meter at a service point.

Manual Meter Type

A business object (BO) used to model properties for meters that are manually read.

Manufacturer

The company that makes devices, defined as an attribute of the device itself.

Market

The jurisdiction or regulatory environment in which a service point participates, defining the valid service providers and their roles. While each service point specifies only one market, different service points throughout the utility's service territory can be linked to different markets.

Market - Fallback Service Provider

For a given market relationship type, a fallback service provider may be defined at the market level, rather than storing the information redundantly on each service point. For example, an entire market might have only one ISO, and if the utility wants to store this information, they can identify the ISO as a fallback service provider for the market and the market relationship type of ISO.

Market - Relationship Type

The valid roles within a market (ISO, Distribution Company, Retailer, etc.) that have some business significance in the application.

Market - Valid Service Provider

The valid service providers for each market relationship type relevant for a given market. The service providers referenced on a service point must be valid for the combination of the service point's market and the market relationship type.

Market Participant

A variety of service provider; a company with a role within a given market such as a retailer or a distribution company.

Measuring Component Consumption Function (MCCF)

An online means of initiating the process of adding or editing measurements. Measuring Component Consumption Functions typically create new initial measurements based on a copy of existing final measurements.

Measurement

A measurement in MDM is synonymous with consumption, which implies that constants or multipliers may have been applied to its value. This term can be used in the context of an IMD or in reference to Final Measurements.

Measurement Condition

Codes that indicate the circumstances (estimated, missing, etc.) of individual measurements. Conditions are assigned to both scalar and interval measurement data both for initial measurement data and final measurements.

Measurement Cycle

The measurement cycle can serve two purposes: it can define the schedule for manual meter reading of devices at service points in that cycle, and it can also be configured to define when to create usage transactions for usage subscriptions associated to service points in the cycle.

Measurement Cycle Route

The route used to collect measurements for a given measurement cycle.

Measurement Cycle Route Sequence

The sequence in which measurements are collected along a measurement route.

Measurement Cycle Schedule

Defines the dates on which devices are scheduled to be read.

Measurement Service

Java services that can be invoked to manipulate interval and scalar measurements. Measurement services are invoked by measurement functions (available through certain zones within MDM), and are also used within processing of usage and VEE rules.

Measuring Component Summary

A zone shown on the VEE Group portal that displays a list of measuring components that reference a given VEE group.

Measuring Component

A single point for which data will be received and stored in the system. A measuring component can be associated to a physical device, which can have one or more measuring components, or it can be stand-alone, meaning that it is not associated to a physical device (for example, an aggregator or interval scratchpad).

Measuring Component Type

The definition of the most important properties of a measuring component, including what it measures, how regularly it measures it, whether it should be connected to a physical device or if it's used as a scratchpad or an aggregator, how its final measurements should be stored and how its user-defined values should be calculated, what rules govern VEE for Measuring Components of the type, as well as numerous display properties that are relevant within MDM. The measuring component type also defines sets of valid attribute values for groups of measuring components belonging to the type.

Measuring Component Types Referencing Group

A zone shown on the VEE Group portal that displays a list of Measuring Component types that reference the VEE group being viewed.

Merge Intervals

Measurement service used to merge a subset of interval data with a specified set of intervals (where overlaps occur, the subset intervals replace the original intervals).

Meter

A device used to measure a quantity of a service (electricity, gas, etc.) delivered to a service point.

Meter Read Download Activity Type

The structure and business rules applicable to downloading meter read information onto a handheld device.

Model

A specific model of a device produced by a manufacturer. Models for a single manufacturer can have diverse service types.

Multiplier

A value that may be applied to adjust the consumption values calculated for a device. Examples include meter/device multiplier, installation constant, loss factor, etc.

Multiply Intervals By Scalar Value

Measurement service that uses the Apply Formula measurement service to multiply the values of a specified set of interval data by a scalar value.

New Scalar Reading

A measuring component consumption function used to create new initial measurement data containing a reading (rather than consumption) for the scalar measuring component displayed in the zone for a user-defined time period. A reading refers to the measurements as read from the meter, while consumption refers to the total consumption, accounting for meter multiplier and/or offset.

Normalized storage

Storing measurement data in a manner that allows for aggregation and reporting of data through database logic (SQL). Applies to both scalar and interval measurements.

Off-Peak Period

A time period during which the least amount of some consumable is being used. OR A period of relatively low system demand as specified by the supplier.

On-Peak Period

A time period during which the greatest quantity of some consumable is being used OR A period of relatively high system demand as specified by the supplier.

One-Way Communication

Communication from head-end system to Oracle Utilities Smart Grid Gateway that does not trigger a response. Examples of one-way communications include usage readings and device events.

Oracle Utilities Meter Data Management

Oracle Utilities application that provides functionality for handling large volumes of meter data to enable increased accuracy, flexibility, and scalability.

Oracle Utilities Smart Grid Gateway

Oracle Utilities application that provides functionality for orchestrating communication with head-end systems to support import of usage and events, and issuing of meter commands.

Outbound Communication

Communication sent from Oracle Utilities Smart Grid Gateway to a head-end system or other external system.

Peak

The maximum value for some measurable quantity recorded over a specified time period. A measuring component that measures peak quantities will record the highest value for the quantity over a period of time.

Peak Demand

The maximum rate of commodity consumption over a specific period of time.

Processing Method

Methods used to define the format or means by which a service provider receives data from the application, such as bill determinants, interval data, or meter events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and usage transactions. Processing methods can also be used to define the information an external system wishes to subscribe to receive from our application. A BO or batch extract code are the typical processing methods defined for the transmission of data to a service provider.

Processing Role

Each processing method has a processing role, which defines the purpose of the processing method. Some examples of processing roles include: * Initial Measurement Creation (D1) * Device Activity Notification (D1) * Usage Transaction Notification (D2) * Usage Transaction Creation (D2)

Profiling of Scalar Data

The process of applying an interval consumption "shape" to a scalar measurement, using an existing interval measuring component. The individual interval values are adjusted such that when totaled, they equal the value of the scalar measurement.

Reader Remark

A type of device event used to capture and/or record specific events or circumstances encountered when a meter reader is manually reading scalar meters.

Reader remarks are submitted with scalar initial measurements when received from a head-end system or meter read collection system. Reader remarks are NOT uploaded along with other device events. Reader remarks are ALWAYS associated with a scalar initial measurement.

Reading

The value recorded by a measuring component at a given point in time. A reading often needs to be interpreted in the context of an earlier reading in order to derive a consumption value that would be stored as a measurement. For example, a reading of 1000 for a subtractive measuring component taken on February 1 in the context of a prior reading of 600 taken on January 15 would result in a consumption (measurement) of 400. Readings can either be consumptive or subtractive.

Register (Measuring Component)

A business object (BO) that represents a scalar register found on a standard or smart meter. It does not have a lifecycle, and should be associated with a device configuration.

Register Type - Physical (Measuring Component Type)

Measuring component type business object (BO) that enumerates the properties used by scalar registers.

Remove Intervals

An initial measurement data function used to remove intervals from initial measurement data. Intervals can be removed from either the beginning or end of the measurement. The end date/time of the measurement is shifted to account for the new number of intervals in the measurement.

Remove Intervals (service)

Measurement service used to remove one or more intervals from a set of interval measurements.

Retail Company

A company that is authorized to buy and re-sell a commodity (such as electricity or gas) directly to customers based on territorial regulations.

Retrieve Interval Consumption

Measurement service used to retrieve one or more interval measurements.

Retrieve Scalar Consumption

Measurement service used to retrieve one or more scalar measurements.

Route Management

A portal used to maintain the sequence of service points within a Measurement Cycle Route.

Save As Interval Consumption

A measuring component consumption function used save all or some of the final measurements shown in the zone as a new initial measurement for a different (new or existing) measuring component.

Scalar Usage

A measurement of the amount of energy, water, gas, etc. consumed for a given measuring component for a given time period.

Seconds Per Interval

Seconds Per Interval, a way of expressing the length of time between which measurements are taken.

Self-Service Meter Read

Meter reads created by end-users via the Oracle Utilities Self-Service application.

Service Investigative Order

Activities created by a service issue monitor when a specified set of events have occurred at a service point. The type of activity created by the service issue monitor is defined on the service issue monitor's type.

Service Issue Monitor

Service tasks that analyze service points to determine if service is needed. If service is determined to be needed, the Service Issue Monitor creates a Service Investigative Order.

Service Order Requests

Requests that orchestrate the field activities (FAs) and smart meter messages (commands) necessary to change the service point and its installation, to enable or disable service, cut service for non-payment, etc.

Service Point

A location at which a company supplies service. Used to store information describing the type of service and how it is measured.

Service Point Identifier Type

Specific types of service point identifiers.

Service Point Identifier

A collection of identifiers for a given service point.

Service Point Parent

The parent of one or more service points.

Service Point Type

A specific type of service point. Defines how the application manages many aspects of the service point's behavior.

Service Provider

External entities that serve various roles relative to the application. These can be a head-end system, a billing system to which the application sends bill determinant data, a market participant in a deregulated environment, an outage management system that receives meter event data from

the application, or other parties that require or provide information to the system.

Service Quantity Identifier

Further distinguishes between measured quantities that have identical UOM/TOU combinations, including situations in which the distinguishing identifier of a UOM is not accurately described as a TOU. SQIs can also be used as a stand-alone representation of a service quantity that is not measured (i.e. one that is not properly described as a UOM) within a Usage SQ collection (e.g. a billing determinant).

Service Task

Records used to capture task-related activities, including tasks performed by users of other Oracle Utilities applications, such as Oracle Utilities Customer Self Service.

Service Type

Specific types of service, such as electric, gas, steam, etc.

Set Condition Codes

An initial measurement data function used to set condition codes within initial measurement data. Condition codes indicate the circumstances (estimated, missing, etc.) of individual measurements. Condition codes are assigned to both scalar and interval measurement data both for initial measurement data and final measurements.

Set Condition

Measurement service used to set the condition (status) code of a specified set of interval data.

Shift Intervals

A function used to shift intervals of initial measurement data forward or backward time. The end result is a measurement with a different start or end time.

Shift Intervals (service)

Measurement service used to shift one or more intervals forward or backward in time.

Smart Meter

A business object (BO) used to model smart meters of different service types.

Smart Meter Installation Event

A business object (BO) that defines the lifecycle and rules for installing a smart meter at a service point.

Smart Meter Type

A business object (BO) for device type that references a head-end system as well as a collection of head-ends that are valid for devices of the type, and indicates whether incoming data incorporates the daylight savings time shift. Additionally, the smart meter type includes a list of valid device configurations for its devices.

Smooth Spikes

An initial measurement data function used to reduce "spike" intervals (intervals with values that are more than a user-defined percentage higher than other intervals within the initial measurement data). The function smooths spikes using linear interpolation as follows:

- Get the value of the interval immediately preceding the spike (the "Left Value").
- Get the value of the interval immediately after the spike (the "Right Value").
- Subtract the Left Value from the Right Value, divide result by two. The result is called the Estimation Adder
- Add the Estimation Adder to the interval immediately before the spike.

Sub Usage Subscription

A "child" usage subscription that is subordinate to another usage subscription (such as when a

utility is issuing a bill for a third-party service).

Substation

A subsidiary station of an electricity generation, transmission and distribution system where voltage is transformed from high to low or the reverse using transformers.

Subtract Scalar Value From Intervals

Measurement service that uses the Apply Formula measurement service to subtract a scalar value from the value of a specified set of interval data.

Subtractive

Describes a measuring component for which consecutive readings must be subtracted to derive a consumption value.

Time of Use

Time of Use - modifiers for a given unit of measure that indicate a period of time during which a quantity has been used, such as On-Peak (meaning during a time when the greatest quantity of some consumable is being used), Off-Peak (meaning during a time when the least amount of some consumable is being used), etc.

TOU Group

A group of TOUs used to limit the set of TOUs usable in a TOU schedule. TOU Groups are used when defining a TOU schedule via a TOU map template.

TOU Map

A collection of TOU map data derived via a given TOU map template at a specific interval size (TOU). A TOU map is typically specified when configuring a usage calculation rule for TOU mapping. This TOU map's data will then be used when summarizing the interval data for each TOU period.

TOU Map Data

An interval date/time and its associated TOU as defined by a TOU map template. For example, if the schedule defined for a TOU map template specifies that the period on weekdays from 9 AM to 5 PM falls into On-Peak, and the data is hourly, rows would be stored in the TOU map data table with the date/time 5/3/2010 at 10 AM, 5/3/2010 at 11 AM, 5/3/2010 at 12 PM, etc., each with a value of On-Peak.

TOU Map Template

The schedule used for TOU map data generation, for example defining year, month, and day ranges and which TOUs should be used during each.

TOU Map Type

Defines certain important properties of TOU maps of the type, including the interval size (SPI) and the valid TOU map templates.

Transformer

A device that transfers electrical energy from one circuit to another.

Two-Way Communication

Communication sent from Oracle Utilities Smart Grid Gateway to an external system, such as a head-end system or edge application that triggers a response. Most commands are two-way communications, where Oracle Utilities Smart Grid Gateway issues a command, and the head-end system sends a response as to the success or failure of the command.

Unit of Measure

Identifies quantities measured, such as KWH, KW, cubic feet, degrees Celsius, etc.

Usage

A generic term for the amount of energy, water, gas, etc. consumed at one or more service points,

sometimes representing quantities that have been adjusted from the original calculated consumption.

Usage Calculation Group

A set of sequenced usage rules used to calculate usage for a usage subscription.

Usage Rule

Business rules / logic used to calculate usage (bill determinants), such as a TOU-mapped consumption calculation. Each rule is a modular unit that can be grouped together and sequenced within a calculation group.

Usage Rule - Apply Math (Interval Data)

A usage rule used to summarize interval measurements and apply a mathematical formula against the results to derive a usage quantity.

Usage Rule - Get Interval Data

A usage rule used to get interval data for a measuring component and date range.

Usage Rule - Get Scalar Details

A usage rule used to assemble scalar readings and measurements (consumption).

Usage Rule - Get TOU Mapped Usage

A usage rule used to summarize interval measurements into TOU buckets based on a TOU map (where the TOU map is created based on the schedule defined within a TOU map template).

Usage Rule - Math

A usage rule used to derive a curve given a formula, apply TOU mapping to a derived curve, and perform mathematical operations on Usage Transaction Service Quantity entries.

Usage Rule - Validate Usage Against Tolerance

A usage rule used to validate calculated usage against a specified tolerance.

Usage Rule Eligibility Criteria

Configured criteria used to determine whether to execute a specific usage rule when calculating usage.

Usage Subscription

A record of an ongoing request to send one or more service points' usage to one or more external systems (such as a billing application).

Usage Subscription Type

A collection of properties defining a class of usage subscriptions. The usage subscription type also controls valid values for various attributes of usage subscriptions.

Usage Transaction

A record of bill determinant calculations for a usage subscription.

Usage Transaction Correction Processor

An activity used to update usage transactions if needed when updated initial measurements are received that might impact the usage transaction(s).

User-Defined Measurement Values

Additional values optionally stored with a given measurement that can be used in various calculations. For example, a customer's gas consumption might be measured in cubic feet, but needs to be sent to a billing system in therms. A user-defined value to convert consumption in cubic feet into therms can be configured, and the therm value will then be stored with the measurement in cubic feet.

Validation, Estimation, and Editing (VEE)

The process by which initial measurement data is validated, estimated (if necessary) and edited (if necessary) based on a set of user-defined rules.

VEE Eligibility Criteria

User-definable conditions that could cause a given VEE rule to be applied or skipped. This could involve the evaluation of some attribute of the device or measuring component, or something else entirely.

VEE Exception

An exception generated during Validation, Estimation and Editing (VEE) processing of initial measurement data. Exceptions are assigned a severity that is used in determining whether or not the initial measurement data should be transitioned into an exception state.

VEE Group

A collection of VEE Rules.

VEE Group Matrix (Factor)

A VEE rule within a VEE group can be configured to pick from a list of VEE groups (referred to as a matrix) whose rules to execute next. This list of VEE groups is configured as the values of a factor. One example of its use could be to call geographically-specific VEE groups from within a larger-purpose group. A residential VEE group might contain a rule that will pick the VEE group to execute based on service point location, where the VEE Group Matrix specifies: SP in the North - VEE Group N SP in the East - VEE Group E SP in the South - VEE Group S

VEE Group Matrix (Factor) Referencing Group

A zone that displays a list of VEE group matrices (factors) that reference the VEE group being viewed in the VEE group portal.

VEE Rule

Standard and custom Validation, Estimation and Editing (VEE) Rules that perform checking and/or manipulation of initial measurement data.

VEE Rule - Duplicate IMD Check

VEE Rule that checks to determine whether the current initial measurement is a duplicate of one already received for the same measuring component.

VEE Rule - Ensure IMD Exists for Sibling MCs

VEE Rule that checks to ensure that initial measurement data exists for siblings measuring components.

VEE Rule - High/Low Check

VEE Rule that compares consumption of incoming initial measurement data against historical data as a means of assuring the reasonableness of the data.

VEE Rule - Interval Adjustment from Scalar

VEE rule that adjusts initial measurement data for interval readings based on an existing scalar value

VEE Rule - Interval Averaging Estimation

VEE rule that estimates missing interval values based on averaging of historical data for the same device and measuring component.

VEE Rule - Interval Interpolation Estimation

VEE rule that estimates missing interval values based on linear interpolation.

VEE Rule - Interval Profile Estimation

VEE rule that estimates missing interval values based on an associated profile measuring component.

VEE Rule - Interval Replacement Rule

VEE Rule that determines whether or not an incoming initial measurement will replace any existing measurements and if so, whether or not to allow this to happen.

VEE Rule - Interval Size Validation

VEE Rule that checks to ensure that the interval size of the incoming initial measurement data matches the value defined in the measuring component type.

VEE Rule - Interval Spike Check

VEE Rule that examines incoming initial measurement data to identify intervals with conspicuously high usage relative to surrounding intervals.

VEE Rule - Multiplier Check

VEE Rule that checks to ensure that the device multiplier value of the incoming initial measurement data matches the multiplier value stored on the measuring component.

VEE Rule - Negative Consumption

VEE Rule that checks for negative consumption.

VEE Rule - Raise Missing Quantity Exception

VEE rule that creates an exception if a specified percentage of measurements lies within the range of condition flag values for missing but not within the range of values for outage.

VEE Rule - Scalar Calculation from Interval

VEE rule that calculates a scalar value from the interval data values from the same date range

VEE Rule - Scalar Estimation

VEE rule that estimates missing scalar readings based on averaging of historical data for the same device and measuring component.

VEE Rule - Scalar Profile Estimation

VEE rule that estimates missing scalar readings based on an associated profile measuring component.

VEE Rule - Scalar Replacement Rule

VEE Rule that identifies whether or not a scalar reading will completely replace an existing measurement and if so, whether or not to allow this to happen.

VEE Rule - Sum Check

VEE Rule that checks the difference between the total consumption of incoming initial measurement data against the total consumption (based on final measurements) for one or more related measuring components on the same device over the same time period.

VEE Rule - Unit Of Measure Check

VEE Rule that checks to ensure that the Unit of Measure (UOM) of the incoming initial measurement data matches the UOM specified on the measuring component's type.

VEE Rule - Zero Consumption Check

VEE rule that creates an exception if it detects zero consumption for the current initial measurement.

Appendix C

Configuration Migration Assistant

The Configuration Migration Assistant (CMA) provides customers with a flexible, extensible facility for migrating their configuration data from one environment to another e.g., from a development environment to a production environment. Data is exported from the source system to a file. The file can then be checked in to a version control system for reuse, or can be immediately imported into the target system and applied.

This appendix describes how the Configuration Migration Assistant can be used with Oracle Utilities products based on the Service and Measurement Data Foundation, including Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway.

Please read and review the **Configuration Migration Assistant** section in the *Oracle Utilities Application Framework Administration Guide* before attempting to use this functionality.

At a high level, migrating data involves the following steps:

1. Create one or more structured **Migration Plans** and specify one or more business objects to migrate.
2. Create a **Migration Request** to specify one or more Migration Plans for export and the specific objects to export using that plan.
3. Create a **Migration Data Set Export** object to register the intent to export specified requests.
4. Create a **Migration Data Set Import** object to register the intent to import completed exports.
5. Review the changes detected by the import and compare process.
6. Apply approved changes.

The following sections provide a reference to the information needed to configure objects and criteria for migration, including the following:

- **Migration Requests**
- **Migration Plans**
- **Wholesale and Piecemeal Migrations**
- **Data that Cannot be Migrated Using Configuration Migration Assistant**

Migration Requests

Migration Requests are unordered lists of Migration Plans that are to be migrated together. Algorithms, SQL statements, or specific keys are used to specify the set of objects you want to export. When complete, the request describes the complete data set that is extracted to the migration export file when the request is executed.

The base package migration request provided can be used as a basis for custom requests. Use the following procedure to access the base package migration request:

1. Navigate to **Admin Menu > Migration > Migration Request**.
2. Enter "**D1**" in the **Migration Request** field.
3. Click **Refresh**.
4. Click the **Description** link in the search results list for the migration request you wish to work with.

Base Package Migration Requests

This section outlines the details of the base package migration requests. Use the **Migration Request** portal to view additional details about each migration request.

Admin Data

- **Migration Request:** D1-AdminData
- **Description:** Admin Data
- **Detailed Description:** This request migrates admin records. This is used for wholesale migrations.
- **Migration Plans:** This migration request includes the following base package migration plans used by Oracle Utilities Service and Measurement Data Foundation and Meter Data Management, including.
 - Service Type wholesale plan (D1-ServiceTypePhysicalBO)
 - Contact Type wholesale plan (D1-ContactTypePhysicalBO)
 - Dynamic Option Type wholesale plan (D1-DynamicOptionTypePhysicalBO)
 - Device Event Type wholesale plan (D1-DvcEventTypePhysicalBO)
 - Exception Type wholesale plan (D1-ExceptionTypePhysicalBO)
 - Manufacturer wholesale plan (D1-ManufacturerPhysicalBO)
 - Unit of Measure wholesale plan (D1-UnitOfMeasurePhysicalBO)
 - SQI wholesale plan (D1-SQIPhysicalBO)
 - Time of Use wholesale plan (D1-TimeOfUsePhysicalBO)
 - Time of Use Group wholesale plan (D1-TOUGroupPhysicalBO)
 - TOU Map Template wholesale plan (D1-TOUMapTmPhysicalBO)
 - TOU Map Type wholesale plan (D1-TOUMapTypePhysicalBO)
 - Service Provider wholesale plan (D1-ServiceProviderPhysicalBO)
 - Processing Method wholesale plan (D1-ProcessingMethodPhysicalBO)
 - Market wholesale plan (D1-MarketPhysicalBO)
 - Measurement Cycle wholesale plan (D1-MeasurementCyclePhysicalBO)
 - Measurement Cycle Route wholesale plan (D1-MeasrmtCycleRoutePhysicalBO)

-
- Measurement Cycle Schedule wholesale plan (D1-MeasrmtCycleSchedPhysicalBO)
 - Factor wholesale plan (D1-FactorPhysicalBO)
 - VEE Group wholesale plan (D1-VEEGroupPhysicalBO)
 - VEE Rule wholesale plan (D1-VEERulePhysicalBO)
 - VEE Eligibility Criteria wholesale plan (D1-VEEEligCritPhysicalBO)
 - Factor Value wholesale plan (D1-FactorValuePhysicalBO)
 - DC Type wholesale plan (D1-DeviceConfigTypePhysicalBO)
 - Device Type wholesale plan (D1-DeviceTypePhysicalBO)
 - SP Type wholesale plan (D1-ServicePointTypePhysicalBO)
 - Usage Group wholesale plan (D1-UsageGroupPhysicalBO)
 - Usage Rule wholesale plan (D1-UsageRulePhysicalBO)
 - Usage Eligibility Criteria wholesale plan (D1-UsageRuleEligCritPhysicalBO)
 - US Type wholesale plan (D1-UsageSubscrTypePhysicalBO)
 - Communication Type wholesale plan (D1-CommunicationTypePhysicalBO)
 - Activity Type wholesale plan (D1-ActivityTypePhysicalBO)
 - MC Type wholesale plan (D1-MeasuringCompTypePhysicalBO)
 - Consumption Extract Type wholesale plan (D1-ConsExtractTypePhysicalBO)
 - Framework Migration plans (this migration request also contains migration plans for several Oracle Utilities Application Framework (F1 owned) objects).

Migration Plans

Migration Plans comprise a group or groups of objects to migrate as a set of related objects. Essentially migration plans sets of instructions describing how the data to be exported is structured. Migration plans allow objects to be migrated together as a logical unit to ensure consistency and completeness.

The base package contains migration plans which can be used as a basis for custom plans. Use the following procedure to access the base package migration plans:

1. Navigate to **Admin Menu > Migration > Migration Plan**.
2. Enter "**D1**" in the Migration Plan field.
3. Click **Refresh**.
4. The **Migration Plan Search** results list displays a list of the base package migration plans:
5. Click the **Description** link in the search results list for the migration plan you wish to work with.

Use the **Migration Plan** portal to view additional details about these migration plans.

Note that base package migration plans cannot be altered. To extend an existing base package migration plan, the base package plan must be duplicated and then altered as needed. The duplicate migration plan can then need to be added to a custom migration request.

Migration Plans for Objects with CLOB-Embedded Links

This section includes suggestions for creating migration plans with CLOB-embedded links.

- When dealing with CLOB-embedded links using a Physical BO, which may not have the specific CLOB elements explicitly defined within its schema, raw SQLs in the Subordinate Instruction's Traversal Criteria can be used to establish relationships involving all CLOB elements (which are saved in the physical database record). However, a separate Subordinate Instruction will have to be created for each SQL corresponding to each element occurrences.
- As an alternative, another option when dealing with CLOB-embedded links is to not use the Physical BO. Instead, use the actual BO which explicitly defines the CLOB elements within its schema. In this case, XPATH Constraints can be used which cover all occurrences of the same element. However, when using this approach, a separate Migration Plan must be created for each such BO. This can become cumbersome depending on the number of BOs associated with any given MO since each of these BOs may have its own set of CLOB elements.
- A third alternative is to create a migration plan that uses the Physical BO as the primary instruction, and then include a subordinate instruction for the real BO, using SQL Traversal to *join the object to itself by its primary key*. The BO will only be included once in the export file, but XPath notation can be used from the subordinate instruction to link to further subordinate objects.

Wholesale and Piecemeal Migrations

There are two general types of migrations used with the Configuration Migration Assistant: wholesale migrations and piecemeal migrations.

Wholesale Migrations

Wholesale migrations are used when migrating all the configuration and/or admin data from one environment to another. For example, a wholesale migration might be used when migrating admin data from a development or test environment to a production environment.

Wholesale migration plans contain Primary Instruction as well as Subordinate Instructions that involve "CLOB-embedded links" (if any exist). Subordinate Instructions for "Standard Constraints" may be included but are deliberately omitted because they are extraneous, bulky and redundant for wholesale migrations. Wholesale migration plans are included in the base package migration request (D1-AdminData) and are provided for use with wholesale migrations.

Executing Wholesale Migrations

This section provides a high-level overview of the steps involved when executing a wholesale migration.

1. Process the "F1-SchemaAdmin" (FW Foundation) migration request (This request contains migration plans for Field, Lookup, Char Type, Currency Code and FK Ref).
2. Process the "D1-AdminData" migration request. This includes copies of framework migration plans (including plans for Business Objects, Algorithms, and Extendable Lookups) from the "F1-FrameworkAdmin" migration request, as well as all of the base package D1-owned wholesale migration plans.
3. Process any of the other delivered framework-based (F1-owned) migration requests as needed (except for the "F1-FrameworkAdmin" migration request which is already incorporated in #2).

All migration requests can be exported at the same time. When importing, Oracle recommends importing, reviewing and applying an entire file/data set before moving on to the next one. If there are objects included in more than one file (which can happen), then two sets of "inserts" will be generated, and only the first will succeed. The second will cause an insert error on that object, and the transaction would be put into "Applied with Error" status. Waiting to perform the import of a second file until after the first is applied means the second dataset will not generate any SQL (since the object is already inserted). When importing all files at once then trying to apply them all, duplicated objects will have to be identified as errors and be marked as "Rejected" before the transaction can be applied.

Piecemeal Migrations

Piecemeal migrations are used when migrating a small portion (or piece) of configuration and/or admin data from one environment to another. For example, a piecemeal migration might be used when migrating VEE groups and rules from a development or test environment to a production environment.

Piecemeal (or non-wholesale) migration plans contain both Primary Instruction as well as all Subordinate Instructions. Subordinate Instructions include "CLOB-embedded links" (if any exist) as well as "Standard Constraints". Piecemeal migration plans are not included in the base package migration request used with wholesale migrations. Base package "non-wholesale" migration plans are provided as examples to help customers migrate data in "piecemeal" or non-wholesale scenarios.

Executing Piecemeal Migrations

This section provides a high-level overview of the steps involved when executing a piecemeal migration.

1. Create one or more piecemeal migration plans as needed based on the specific configuration/admin data to be migrated. (the base package contains a number of sample "non-wholesale" migration plans that be used as examples for this).
2. Create a custom migration request that contains the migration plans created in step 1, and specifies the keys of the objects to be migrated. Keys can be specified using any of the selection types - SQL, algorithm or specific key values. Be sure to include all primary key values for the objects.
3. Process the custom migration request (export/import).

When importing piecemeal migrations, Oracle recommends importing, reviewing and applying an entire file/data set before moving on to the next one (similar to the steps recommended for wholesale migrations).

Base Package Piecemeal Migration Plans

The base package include the following piecemeal (non-wholesale) migration plans. These are provided as examples for customers when creating their own piecemeal migration plans.

Migration Plan	Description	Primary Instruction Business Object
D1-MeasuringComponentTypePlus	MC Type Non-wholesale plan	D1-MeasuringCompTypePhysicalBO
D1-RegisterTypePlus	Register Type Non-wholesale plan	D1-RegisterTypePhysical
D1-UsageGroupPlus	Usage Group Non-wholesale plan	D1-UsageGroupPhysicalBO

Data that Cannot be Migrated Using Configuration Migration Assistant

This section provides details regarding data that cannot be migrated using the Configuration Migration Assistant.

- **Key Type**
- **Links to System Generated IDs**

Key Type

Only data with specific types of keys can be migrated using the Configuration Migration Assistant. To be included in a base package migration, the “Primary” table for the object’s maintenance object main must meet the following criteria:

- `KEY_TYPE_FLG = 'USR'` (User-defined).

Note: `KEY_TYPE_FLG` is a column in `CL_MD_TBL`.

Because they all use system-generated keys, the following maintenance objects are not supported by Configuration Migration Assistant:

- Dynamic Option (D1-DOP)
- Dynamic option event (D1-DOPEVT)
- Time of Use Map (D1-TOUMAP)

Data based on these objects must be migrated manually between environments.

Links to System Generated IDs

Data with links to system-generated IDs are not supported in the Configuration Migration Assistant.

The Usage Rule and Measuring Component Type BO schemas contain CLOB elements that link to measuring components or TOU Maps via system-generated IDs.

For example:

- Usage Rule BO schema - contains 'measuringComponentId' and 'touMapId' elements
- Measuring Component Type BO schema - contains 'defaultTOUMapForDisplay' element

Configuration Migration Assistant does not currently support migration of objects with system-generated IDs. As of this release, the associated measuring component type or TOU map data needs to be migrated manually.

Appendix D

Information Lifecycle Management

This chapter describes how the Information Lifecycle Management (ILM) feature of the Oracle database can be used with Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway.

- **ILM Components**
- **Archive Eligibility**

Before you read this chapter, you should familiarize yourself with how Information Lifecycle Management works within the Oracle Utilities Application Framework.

- Refer to the **Information Lifecycle Management** section in *Chapter 5: Database Tools* in the *Oracle Utilities Application Framework Administration Guide* for general information about how ILM works with Oracle Utilities Application Framework applications.

ILM Components

This section provides an overview of the components used by the Information Lifecycle Management functionality, including:

- **ILM-Enabled Maintenance Objects**
- **Maintenance Object and Business Object Options**
- **Retention Periods**
- **Archive Eligibility**

ILM-Enabled Maintenance Objects

The **ILM Configuration** master configuration contains a list of the maintenance objects configured to support ILM. The ILM Managed Maintenance Object section displays each maintenance object as well the related eligibility algorithms and batch control crawler for each.

Maintenance Object and Business Object Options

The ILM functionality in Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway makes use of some specific maintenance object and business object options. These options are used by algorithms that determine the archiving eligibility of different types of data. In particular, these options can be used to require that a transaction be in a final state before it is considered eligible for archiving. For instance, if an implementation wants all transactions of a certain maintenance object to be final before archiving, the maintenance object level options can support that. If the same implementation only wants certain types of transactions for maintenance object to be final then they can set it at the business object level.

Support for archiving of non-final data is provided because data retention periods are most often measured in years, and there can be any number of business cases in which data could exist in a non-final state for many years and as such be considered eligible for archiving.

Refer to the Detailed Descriptions of the maintenance object's ILM eligibility algorithms and **Evaluating Maintenance Object and Business Object Options** on page D-3 for more information about how these options are used.

Maintenance Object Option Types

In addition to the "ILM Crawler Batch Control" and "ILM Retention Period in Days" maintenance object option types, the following maintenance option types are available, but not configured in the base package:

- **ILM Final Status Field Value:** If the maintenance object is configured to support ILM, this option can be used by the maintenance object's ILM eligibility algorithm to define an explicit state that a record should be in to be eligible for archiving. Multiple entries for this option type can be added if the maintenance object has more than one state value that should be considered.
- **ILM Restrict By BO Final Status (Y/N):** If the maintenance object is configured to support ILM, this option can be used by the maintenance object's ILM eligibility algorithm to potentially consider a business object's status value when evaluating ILM eligibility. Valid values are "Y" and "N".
- **ILM Restrict By Status (Y/N):** If the maintenance object is configured to support ILM, this option is available for use by the maintenance object's ILM eligibility algorithm to potentially consider a record's status value when evaluating ILM eligibility. Valid values are "Y" and "N".

Note: This parameter must be set to "Y" in order for the "ILM Restrict By BO Final Status" and or the "Final Status Required for Archive" (see below) options to be considered.

Business Object Option Types

The following business object option type is available, but not configured in the base package:

- **Final Status Required for Archive (Y/N):** If the business object's maintenance object is configured to support ILM, this option is available for use by the maintenance object's ILM eligibility algorithm to potentially consider the status value when evaluating ILM eligibility. Valid values are "Y" and "N".
- Note: This option takes precedence over the "ILM Restrict by BO Final Status (Y/N)" maintenance object option.

Evaluating Maintenance Object and Business Object Options

The eligibility algorithms retrieve the values for the "ILM Restrict By Status", "ILM Restrict By BO Final Status", and the "BO Option for Final Status Required for Archive" options to evaluate if records must be in a "Final" status in order to archived.

Only when the "ILM Restrict By Status" option is set to "Y" will it be possible that the transaction must be final. The "BO Option for Final Status Required for Archive" will take precedence over the "ILM Restrict By BO Final Status" option. The table below shows the list of possible combinations and whether or not that combination means the transaction must be in a final status.

Maintenance Object: ILM Restrict By Status	Maintenance Object: ILM Restrict By BO Final Status	Business Object: Final Status Required for Archive	Transaction Must be In Final Status
N	Y	Y	N
N	Y	N	N
N	N	Y	N
Y	N	N	N
Y	Y	N	N
Y	Y	Y	Y
Y	N	Y	Y
Y	Y	Not provided	Y
Y	N	Not provided	N

Archive Eligibility

This section describes details related to the ILM eligibility for each of the maintenance objects which support ILM. This includes:

- **Retention Periods**
- **Eligibility Summary**
- **Eligibility Algorithms**

Retention Periods

Retention periods define the number of days that records should remain active. Each record created for an ILM managed maintenance object is assigned an ILM date (generally set to the system date) and an ILM archive switch (initially set to N). A record is considered active as long as its ILM date plus the retention period is not in the past. ILM crawlers select records with a retention period that has elapsed; these records are considered eligible for archival and the ILM archive switch may be set accordingly by an eligibility algorithm. Note that the default retention period is only used if an ILM managed maintenance object does not have a retention period maintenance object option defined.

Sub-Retention Periods

Sub-retention periods are used for those types of transactions that should be retained for less time than the Default Retention Days for the transaction's maintenance object. For example you may decide to retain scalar measurements for a shorter time than interval measurements. In this case, you would define a shorter duration for the Scalar IMD Retention Days parameter on the master configuration.

For records for which a sub retention period has been defined, the ILM date will be adjusted backwards to account for the shorter period to be retained. For example, if a transaction was added on 7/31/2014 and the default retention period was 5 years (1825 days), and the sub retention period for that particular type of data was 3 years (1095 days), then the ILM date would be set to 7/31/2012, calculated as follows: $\text{ILM Date} = \text{Current Date} - (\text{Default Retention Period} (1825) - \text{Sub Retention Period} (1095))$.

Retention Periods and ILM Dates

By default, the ILM date is calculated as follows: $\text{ILM Date} = \text{Current Date} - \text{Sub-Retention Period}$. If the "ILM Retention Period in Days" maintenance option is defined for a particular maintenance object, that is used as the retention period. This can be defined in either the ILM Master Configuration or on the maintenance object Options tab. If this option is not defined, the Default Retention Days from the ILM Master Configuration record is used.

Master Configuration

The ILM Configuration master configuration is used to define the retention and sub-retention periods used by ILM for Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway. This master configuration defines the following options.

Default Retention Days: defines the number of days that records should remain active.

Sub-Retention Periods: sections that define sub-retention periods for:

- Activities
- Device Events
- Initial Measurements (IMD)

Note that the Default Retention Days for Activity, Initial Measurement Data, and Device Event Retention specified on the master configuration are persisted in the "ILM Retention Period in Days" option on the corresponding maintenance object.

Refer to the embedded help on the master configuration for more information about these options.

Eligibility Summary

The table below summarizes details of the eligibility algorithms for each maintenance object supported by ILM. Information on this table includes:

- **Archived By:** Indicates how archiving for transactions for the maintenance object is initiated. For example, device events can be archived by themselves or as part of a related activity.
- **ILM Date Computation:** Indicates the type(s) of transactions that impact the calculation of the ILM Date for transactions for the maintenance object. "Itself" indicates that the ILM date is not impacted by other transactions.
- **Specific Eligibility Considerations:** Indicates specific considerations used by the eligibility algorithm.
- **Archived Transactions:** Indicates other transactions that are archived when transactions for the maintenance object are archived. For example, VEE exceptions are archived with initial measurements.

Maintenance Object	Archived By	ILM Date Computation	Specific Eligibility Considerations	Archived Transactions
Initial Measurement	Itself	Itself	VEE Exceptions must be eligible Associated Related MC Synchronization activity must be complete	VEE Exceptions
VEE Exception	Itself	Earliest of Initial Measurement or VEE Exception	Related Service Issue Monitor must be complete	
Usage Transaction	Itself	The latest ILM Date for all related usage transactions	All related usage transactions must be eligible for archiving	Related Usage Transactions
Device Event	Itself Activity	Itself (for stand-alone events) Latest ILM Date for paired events	Paired Events must be eligible Related Service Issue Monitor must be complete Related activity must be complete	
Activity	Itself Parent Activity	Superseded by the latest ILM date of the following child objects: Device Events Activities Completion Events	Child objects must be eligible Command requests must not be associated to a non-final Service Issue Monitor	Child Activities Device Events Outbound Communications Inbound Communications Completion Events
Outbound Communication	Activity	The earlier of its ILM Date or that of the initiating activity's ILM Date		Inbound Communication
Inbound Communication	Itself Outbound Communication	Itself (if no related outbound communication) Superseded by related outbound communication		Device Events

Eligibility Algorithms

ILM eligibility algorithms are provided in the base package for use with Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway. These algorithms are defined for the "ILM Eligibility" system event on their respective maintenance objects. Refer to the Detailed Descriptions of the maintenance object's ILM eligibility algorithms for information about how these algorithms determine ILM eligibility.

In the case of all of the eligibility algorithms, when a transaction is found to be ineligible, the eligibility algorithms set the ILM Date for that transaction to the current date.

The following section provides additional information beyond that provided in the algorithms' Detailed Descriptions.

Activities

The "ILM Eligibility - Activity" algorithm determines the archiving eligibility of activities. The following sections provide additional information about how the "ILM Eligibility - Activity" algorithm determines the archiving eligibility of specific types of activities.

Request Orchestration Activities

Request orchestrations also archive any child activities that were created, and those child activities archive any child activities or data (completion events, device events, etc.).

Field Activities / Command Activities

Field activities and command request activities are archived either as part of a request orchestration or by themselves. When a field activity/command request is archived, it also archives any of the following child transactions:

- Update/Cancel Orchestrators
- Communication Out
- Communication In
- Completion Event

Non-Dispatchable Activities

Non-Dispatchable activities archived either as part of a request orchestration or by themselves. When a non-dispatchable activity is archived it also archives child completion events:

Orchestration Maintenance Activities

Orchestration Maintenance activities are only archived by themselves if they do not have a related activity. When an orchestration maintenance activity is archived it also archives any of the following child transactions:

- Communication Out
- Communication In

Device Event Activities

Presently device event activities are limited to outages which have an initiating and an ending device event. These types of activities cannot be archived until their related device events are either archived or ready to archive. However the device events do not have to wait for the activity to archive so the device events are not updated with the activity's ILM date.

Bulk Activities

Bulk Activities are comprised of the Bulk Header and the Bulk Request/Response, and depending on how the header was created there will be one header to many bulk requests/responses. The Bulk Header will only be eligible for archive if all related Bulk Request activities are also eligible for archiving. All related bulk activities are archived together. Bulk activities also generate one to many command request activities. Those individual command activities are archived separately.

Payload Statistics Activities

Payload Statistics activities can either be the Payload Summary record or the Payload Statistic record. Eligibility is based on the Payload Statistics activity, which is only be eligible for archiving if all the related Payload Summary and Payload Error activities are also eligible for archiving. These activities are only archived by themselves if they do not have a related activity. Otherwise both the payload statistics and the payload summary activities are archived at the same time.

Extract Request Activities

These activities are used to request data from the head-end system on a periodic basis. Extract request activities should be in a final state prior to being archived.

Other Activities

Other types of activities can be archived provided they do not require any special logic for handling for archiving purposes, such as checking for related data.

These activity types can archive without checking related transactions:

- Consumption Sync
- Dimension Scanner
- Error Activity
- Measurement Quantity (deprecated)
- Meter Read Download Activity
- Suppression
- Usage Transaction Correction Processor

In the event that your implementation uses custom activity types that require special handling, a custom algorithm should be created and added prior to the base package algorithm to preemptively handle the activity type category.

Appendix E

Multiple Time Zone Support

This chapter describes support for maintaining data in multiple time zones, including information about the application behaves differently when the multiple time zone functionality is enabled. This includes the following:

- **Overview**
- **Terms and Definitions**
- **Enabling Multiple Time Zone Support**
- **Master Data**
- **Data Synchronization**
- **Initial Measurement, Device Event, Usage Transaction Import**
- **VEE Rules and Processing**
- **TOU Map Data Generation**
- **Scalar Periodic Estimation**
- **Outbound Messages**
- **Aggregations**
- **Smart Grid Gateway Adapters**
- **Online Creation of Initial Measurements**
- **Implementing Multiple Time Zones**

Overview

Some utilities have operations in more than a single time zone. When operating in multiple time zones, Oracle Utilities Meter Data Management and Smart Grid Gateway provide the following high level functionality:

- The ability to receive data in any time zone and convert it to the time zone of the device
- The ability to define what time zone a measuring component, device configuration, service point, or usage subscription resides in
- The ability to aggregate data across time zones into a single time zone or without respect to a time zone (see **Aggregations** on page E-17 for more information).
- The ability to export data to an external system with the appropriate time zone information to enable the external system to consume data from different service points in different time zones

Date/Time Storage and Display

All date/times in the system will be stored and displayed in the application installation time zone. Date/times are stored in Standard Time, but are displayed in "Wall Time" (see **Terms and Definitions** on page E-3 for descriptions of Standard Time and Wall Time).

Terms and Definitions

This section outlines a number of terms used when describing the multiple time zone feature.

- **Daylight Saving Time (DST):** A period of the year where clocks are adjusted forward or backward (typically by one hour). This results in a skipped period of time in the spring (when clocks are adjusted 1 hour forward) and a duplicate period of time in the fall (when clocks are adjusted 1 hour backward) In the US this shift occurs at 2AM.
- **Standard Time:** The time in a given time zone without any DST shift applied
- **Legal Time:** The time in a given time zone with DST shift applied. Note that Legal and Standard Time are the same during non-DST periods.
- **Wall Time:** The time that appears on a clock on the wall, which is the standard time with the DST shift applied (i.e. same as legal time).
- **Universal Coordinated Time (UTC):** The international standard time zone - all other time zones are expressed as an offset to this time zone.
- **Greenwich Mean Time (GMT):** The time as determined by the “average” day at the Greenwich observatory. The only difference between GMT and UTC are GMT’s disregard for the fluctuations in the normal earth-sun interaction. For practical purposes GMT and UTC can be considered to be equal.
- **Local Time:** Time is typically captured in the time zone of the physical location. For example, an office San Francisco has a local time of Pacific (PST), while an office in Boston has a local time of Eastern (EST). This is used to describe a way of requesting data without regard for a given time zone. If data is requested for 1:00 PM for offices in Boston and San Francisco in local time, it means the request would want to add data from the 1:00 PM EST (UTC-5) local time and the 1:00 PM PST (UTC -8) local time together.
- **Absolute Time:** This refers to a request for data that would want to retrieve data with respect to a given time zone. This means that all data from disparate time zones must be converted to a single time zone and combined. If data for 3:00 PM EST is requested in absolute time it means the request would want to add 3PM EST (UTC -5) to 12PM PST (UTC -8). The specific time zone absolute time represents will depend on the request.
- **Shifting Date/Times:** Typically "shifting" refers to moving a date/time from wall time to standard time or standard time to wall time.
- **XSD Date/Time Format:** A standard XML format used to define date/times. This format is notated as follows:

CCYY-MM-DDThh:mm:ss

Example: 2015-05-04T21:32:52

This format can also support identifying the time zone, by including "Z|(+) | -)hh:mm" following the date/time.

Example: 2015-05-04T11:35:55Z+5

- **OAAF Date/Time Format:** The format used to define date/times within Oracle Utilities Application Framework applications. This format is notated as follows:

YYYY-MM-DD-HH24-MM-SS

Example: 2015-05-04-1124-35-55

Note: This format does not support time zone identification.

- **Installation Time Zone:** The time zone defined for the implementation. This is defined on the **Main** tab of the **Installation Options - Framework** portal.

Enabling Multiple Time Zone Support

Multiple time zone support is enabled via the **Multi Time Zone Support** option on the **General System Configuration** feature configuration.

Valid values for the **Multi Time Zone Support** option are:

- D1YS (Yes)
- D1NO (No)

The value defined for this option is used within algorithms to determine the logic to execute when supporting a single time zone or multiple time zones.

The following sections of this chapter provide additional details regarding how this Feature Configuration impacts the application.

Refer to the **Defining Feature Configurations** section in the *Oracle Utilities Application Framework Administration Guide* for more information about defining feature configuration options.

Master Data

This section outlines the impact of multiple time zone support on master data such as device configurations, service points, usage subscriptions, and others. Specifically, this section defines how time zone validation is performed for these objects.

Overview

When multi time zone support is disabled, validation algorithms ensure that the time zone of several master data objects must match the installation time zone.

When multiple time zone support is turned on, the time zone field on master data can be set to any valid time zone defined in the system. However, the same validation algorithms are used to ensure that the entire device hierarchy is on the same time zone.

- There is no ability to mix time zones between a device configuration, service point, usage subscription, and measuring component.
- For example a device configuration in US/Central installed at a service point in US/Central cannot have its time zone changed because it cannot be in conflict with the service point.
- In this scenario the service point time zone can be changed and it will also change the device configuration time zone (so long as there are no conflicts created with other service points that the device configuration has been installed for or Usage Subscriptions that the service point shares with other service points)

The sections below outline how multiple time zone validations are performed for different master data objects. Refer to the Detailed Description on the Algorithm Type for each these algorithms for more details.

Device Configurations

Validation Algorithm: Validate Device Configuration Time Zone (D1-VALDCTMZN)

When the Multi Time Zone Support option is not configured or set to "No" then the time zone of the Device Configuration must match the installation time zone.

If the Multiple Time Zone Support option is configured as "Yes" then the Device Configuration's time zone can be set to any time zone. However, the time zone can only be changed when it is not currently installed at a Service Point. When a Device Configuration is installed at a Service Point the Service Point will be the single point for changing the Time Zone.

Service Points

Validation Algorithm: Validate Service Point Time Zone (D1-VALSPTMZN)

Post-Processing Algorithm: Synchronize SP Time Zones (D1-SYNSPTMZN)

When the Multi Time Zone Support option is not configured or set to "No" then the time zone of the Service Point must match the installation time zone.

If the Multiple Time Zone Support option is configured as "Yes" then the Service Point's time zone can be set to any time zone. However, if the Service Point has Device Configurations installed (past or present) or Usage Subscriptions that reference the Service Point in the Usage Subscription Service Point list then further validation is necessary for the time zone change.:

- If the Service Point is a child Service Point to another service point the time zone cannot be changed since all child Service Points must have the same time zone as their parent.
- The Service Point cannot have any Usage Subscriptions that have other Service Points or Direct Channel Measuring Components, because all Service Points and Measuring Components related to a Usage Subscription must have the same time zone.

- The service point cannot have any Device Configurations that have, or are, installed at a different Service Point, because a Device Configuration must always have the same time zone as the Service Point where it has been installed (past or present).

If the time zone change passes validation and either a child Service Point, a Device Configuration, or a Usage Subscription has a time zone that differs from the Service Point a warning will be issued to the user indicating that those objects will have their time zones updated as well.

The "Synchronize SP Time Zones" post processing algorithm will perform any necessary updates to the associated child Service Points, Device Configurations, or Usage Subscriptions.

Usage Subscriptions

Validation Algorithm: Validate Usage Subscription Time Zone (D2-VALUSTMZN)

When the Multi Time Zone Support option is not configured or set to "No" then the time zone of the Usage Subscription must match the installation time zone.

If the Multiple Time Zone Support option is configured as "Yes" then the validation algorithm prevents a time zone being defined for a Usage Subscription that does not match the time zone of either the Service Points in the Usage Subscription Service Point list or the Measuring Components in the Direct Channel Measuring Components list. It also validates that all Service Points and Measuring Components have the same time zone. This means that each Measuring Component identified as a Direct Channel on the Usage Subscription must have a time zone defined.

Install Events

Validation Algorithm: Validate Service Point and Device Configuration Time Zones (D1-VALSPDCTZ)

When the Multi Time Zone Support option is not configured or set to "No" then the validation algorithm will exit since the Device Configuration and Service Point will both have the installation time zone.

If the Multiple Time Zone Support option is configured as "Yes" then the validation algorithm will ensure that the Service Point and the Device Configuration have the same time zone. If the Service Point and the Device Configuration have different time zones but the Device Configuration is not installed at any other service points, an error message will instruct the user to update the Device Configuration's time zone prior to installing. If the Device Configuration has been installed at other Service Points, an error will indicate that the Device Configuration cannot be installed at that Service Point and a new Device Configuration will need to be created with a different time zone.

Data Synchronization

Data synchronization of master data is important for any implementation that is supporting multiple time zones. Master data moving from a customer information system (or CIS such as Oracle Utilities Customer Care and Billing) to Oracle Utilities Meter Data Management needs to be defined in the appropriate time zone within Oracle Utilities Meter Data Management to ensure that the meter data is correctly processed. Note that the master data synchronization process does not perform any date/time conversion, and assumes that any dates sent from the external systems are represented in local time of the device.

If the data coming from the CIS includes time zone information, the data synchronization business objects algorithms correctly set the appropriate time zone in the master data object in Oracle Utilities Meter Data Management. In the event that the data coming from the CIS does not include the time zone, the data synchronization algorithms can either populate the time zone by default (based on the Installation Time Zone) or can return an error.

The "Validate and Default Sync Request Inbound Time Zone" algorithm (D1-VALDFSITZ) defaults the time zone element from the time zone specified in the Installation Options. If the Multiple Time Zone Support option is configured as "Yes" this algorithm will first check if the target time zone element is populated, if so, it will skip setting the time zone.

If the "Error if Time Zone Not Populated" parameter is set to "Yes", the time zone must be populated for the synchronization object. If the time zone is not populated, a log entry will be created and the business object will be transitioned to the "Validation Error" status. This parameter is only applicable when the Multiple Time Zone Support option is configured as "Yes."

Initial Measurement, Device Event, Usage Transaction Import

This section outlines the behavior of the initial measurement, device event, and usage transaction import processes when using the multiple time zone functionality.

Head-End System Service Providers - Defining the Date/Time Format

How a head-end system sends date/time information plays a large role in how that data is processed. This is because different date/time formats are handled differently by the XAI inbound services. If an incoming payload to an XAI inbound service contains date/times in the XSD format with a UTC offset, the XAI inbound service will perform time zone conversion on those dates/times to the appropriate installation time zone (either wall time or standard based on the schema configuration). However, if the date/times are in any other format, the XAI inbound services will not perform time zone conversion.

The Head-End System service provider business object (D1-HeadEndSystem) contains the following fields that can be used to specify how the head-end system sends date/time information.

Field	Valid Values
IMD Import Date/Time Format	<ul style="list-style-type: none"> With Time Zone Without Time Zone (Default behavior)
Event Data Import Date/Time Format	<ul style="list-style-type: none"> With Time Zone Without Time Zone (Default behavior)
Command Response Date/Time Format	<ul style="list-style-type: none"> With Time Zone Without Time Zone (Default behavior)

The IMD Seeder and Device Event Seeder business objects contain logic that will shift date/times from local wall time to local standard time as well as convert from one time zone to another.

How this logic proceeds is determined by how the head-end formats its date/times, which can be specified using the “IMD Import Date/Time Format” and “Event Data Import Date/Time Format” options (for initial measurements and device events, respectively).

Value	Behavior
With Time Zone	<p>Used when the head-end system sends date/times in the full XSD format (with the UTC offset).</p> <p>Example: 2015-05-04T11:35:55Z+5</p> <p>The logic assumes the date/times are in installation time zone standard.</p>

Value	Behavior
Without Time Zone (Default behavior)	<p>Used when the head-end system sends date/times in any other non-XSD format or when an XSD date/time does not include a UTC offset.</p> <p>Example: 2015-05-04T11:35:55</p> <p>The logic shifts the data based on the device-level data shift information as well as the time zone defined on the service point (or device configuration if no installation exists)</p>

IMD Seeder Behavior

The IMD Seeder business object (D1-IMDSeeder) performs differently when the multiple time zone functionality is enabled. This section outlines this behavior.

Time Zone Conversions

When converting initial measurement date/times from the time zone of the meter/head end to the time zone of the installation, the conversion algorithms can retrieve a non-install time zone from the device to use as the source time zone. All time zone conversions have a target time zone as the installation time zone (see **Time Zone Translation and Conversion** on page 6-3 for information about the impact of the "Incoming Data Shift Flag" on time zone conversions).

The "Perform Date/Time Adjustments and Undercount/Overcount Check" algorithm (D1-DODTTMADJ) performs any time zone conversion that may be necessary. If the source time zone of the incoming data differs from the target time zone then the date times will be converted to the target time zone.

The source time zone is determined as follows:

1. The time zone element in the "preVEE" group
2. The translated version of the external time zone in the "preVEE" group
3. The Service Point time zone
4. The Device Configuration time zone
5. The Measuring Component time zone
6. The installation time zone.

Note: if the head-end system is configured to send date/times with time zone information, the source time zone will be the installation time zone because the XAI inbound service will have performed a time zone conversion to the installation time zone prior to loading the data into the application.

The target time zone is set to the installation time zone.

Installation and Time Zone Validation

The "Derive Service Provider and Measuring Component" algorithm (D1-DER-SPRMC) can prevent initial measurements from being processed for devices that are not installed where the incoming initial measurement does not have any time zone indications. This is to ensure the time zone is correct and the time zone on the device configuration may not be correct until it has been installed at a service point. For example, a device configuration might have an incorrect time zone

prior to installation at a service point if it was successfully synchronized but the service point failed to synchronize for some reason. This validation can be turned off through the "Error if MC is Not Installed" algorithm parameter.

If the "Error if MC is Not Installed" parameter is set to "true," the initial measurement will be transitioned into an error state if the following criteria are met:

- The Multiple Time Zone Support option is configured as "Yes"
- The Device Configuration is not installed at a Service Point
- The Device's Head-End system sends initial measurement data without time zone information (i.e. it does not use XSD format with a supplied UTC offset)
- The initial measurement does not contain a time zone or external time zone value

Device Event Seeder

The Device Event Seeder business object (D1-DeviceEventSeeder) performs differently when the multiple time zone functionality is enabled. This section outlines this behavior.

Time Zone Conversion

When converting device event date/times from the time zone of the meter/head end to the time zone of the installation can retrieve a non-install time zone from the device to use as the source time zone. All time zone conversions have a target time zone as the installation time zone.

The "Shift Event Date/Times to Standard" algorithm (D1-SHEVTDITM) can perform both date/time shifting from local wall time to local standard time as well as time zone conversion from a source to target time zone.

The source time zone is determined as follows:

1. The time zone element
2. The translated version of the external time zone element
3. The Service Point time zone
4. The Device Configuration time zone
5. The Measuring Component time zone
6. The installation time zone.

The target time zone is set to the installation time zone.

Installation and Time Zone Validation

The "Shift Event Date/Times to Standard" algorithm (D1-SHEVTDITM) can prevent device events from being processed for devices that are not installed where the incoming device event does not have any time zone indications. This is to ensure the time zone is correct and the time zone on the device configuration may not be correct until it has been installed at a service point. This validation can be turned off through the "Error if Device is Not Installed" algorithm parameter.

If the 'Error if Device is Not Installed' parameter is set to "Yes," the device event will be transitioned into an error state if the following criteria are met:

- The Multiple Time Zone Support option is configured as "Yes"
- The Device's Device Configuration is not installed at a Service Point
- The Device's Head End system sends Event data without time zone information (i.e. it does not use XSD format with a supplied UTC offset)
- The Device Event does not contain a time zone or external time zone value

Usage Transaction Seeder

The Usage Transaction Seeder business object (D2-UsageTranSeeder) performs differently when the multiple time zone functionality is enabled. This section outlines this behavior.

Time Zone Conversion

The "Perform Time Zone Conversion" algorithm (D2-PERUTTZCV) supports the conversion of time zones for incoming usage transactions for implementations that support multiple time zones. This algorithm will exit without processing if any of the following are true:

- The Multi Time Zone Support option is not configured or set to "No"
- The incoming seeder payload does not contain a service provider.
- The time zone conversion has already taken place (the Process Data Is Time Zone Converted element has been set to yes)

This algorithm performs time zone conversion by identifying the source time zone of the seeder information and converting the date/times to the target time zone of the usage transaction.

The source time zone of the seeder information is determined as:

1. The OUAF time zone from the Time Zone element of the payload
2. The translated version of the External Time Zone from the payload
3. The Usage Transaction's Usage Subscription's Time Zone
4. The Installation Time Zone

The target time zone of the usage transaction is the installation time zone

Note: if the external application is configured to send date/times with time zone information, the source time zone will be the installation time zone because the XAI inbound service will have performed a time zone conversion to the installation time zone prior to loading the data into the application.

External Application Service Providers - Defining the Date/Time Format

The external application service provider business object (D1-ExternalApplication) contains the following field that can be used to specify how the external system sends date/time information

Field	Valid Values
Usage Transaction Import Date/Time Format	<ul style="list-style-type: none"> • With Time Zone Used when the external system sends date/times in the full XSD format (with the UTC offset). • Without Time Zone (Default behavior) Used when the external system sends date/times in any other non-XSD format.

VEE Rules and Processing

VEE processing can be impacted when the multiple time zone functionality is used. This section outlines how VEE processing behaves when multiple time zones are enabled.

Factor Processing

Factor effective dates do not have a time zone associated to them and are assumed to be in the local time zone of the data being validated. For example a factor effective 4/1/2015 12AM would be considered to be effective 4/1/2015 12AM US/Pacific for a San Francisco meter and 4/1/2015 12AM US/Eastern for a New York meter.

The Factor processor retrieves factor values based on a set of inputs including a start and end date. The processor returns a list of factors effective for the period of time covered by the start to end period date/times. The factor processor performs date/time conversion to the start and end date/times when retrieving the factor values based on an effective date/time.

When converting factor value effective date/times:

- The source time zone is the installation time.
- The target time zone is one of the following, (in order)
 1. The Usage Subscription time zone (if a usage subscription is populated)
 2. The Service Point time zone
 3. The Device Configuration time zone
 4. The Measuring Component time zone

If the time zone retrieved is different than the installation time zone then the date/times are converted to the target time zone, and the converted date/times are used to retrieve the effective dated factor value.

Holidays

Holiday effective dates do not have a time zone associated to them and are assumed to be in the local time zone of the data being validated. For example the 4th on July 7/4/2015 would be considered to be effective beginning 7/4/2015 12AM US/Pacific (-8 GMT) for a San Francisco meter and 7/4/2015 12AM US/Eastern (-5 GMT) for a New York meter

To evaluate whether a given date/time falls on a holiday, the date/time being evaluated must be converted to wall time in the time zone of the device.

For example assume the following setup: the installation time zone is Eastern (-5 GMT) and the device is in Pacific (-8 GMT) and a given holiday is defined from 12/25/2014 00:00:00 to 12/26/2014 00:00:00. The device's intervals are stored in Eastern, but should be evaluated based on the holiday's start and end date/times in Pacific. To do so the interval date/times will be converted from install standard to wall time of the device's time zone.

Here are how some intervals from the Pacific device would be evaluated (note these are in Eastern time as they would be stored):

- 12/25/2014 00:00:00 - Not on the holiday (converted to 12/24/2014 21:00:00)
- 12/25/2014 04:00:00 - On the holiday (converted to 12/25/2014 01:00:00)
- 12/26/2014 00:00:00 - On the holiday (converted to 12/25/2014 21:00:00)
- 12/26/2014 04:00:00 - Not on the holiday (converted to 12/26/2014 01:00:00)

The Interval Averaging VEE rule also takes advantage of the above holiday and business calendar logic.

TOU Map Data Generation

This section outlines the impact of multiple time zone support on time of use data such as TOU Map Types, TOU Maps, and Dynamic Option Types.

Overview

When the Multiple Time Zone Support option is configured as "Yes":

- The time zone field on TOU Map, TOU Map Type, and Dynamic Option Types can be set to any valid time zone defined in the system.
- TOU Map data generated for non-installation times zone will be stored in the installation time zone, but the transitions between TOU periods will be based on the time zone of the TOU Map.

Because measurement data is stored in the installation time zone, storing TOU map data in the same way allows the TOU map data to be compared to the measurement data directly without any translation.

For example a TOU Map with a time zone of US/Pacific (-8 GMT) with a US/Eastern (-5 GMT) installation time zone that transitions to "On Peak" at 4PM will store that transition as 7PM (which is 4PM adjusted ahead three hours to match the installation time zone of US/Eastern)

Validations

This section outlines how time zone-related validations are performed for TOU-related objects.

TOU Map Type

Validation Algorithm: Validate Time of Use Map Type Time Zone (D2-VALTMTYT'Z)

When the Multi Time Zone Support option is not configured or set to "No" then the time zone of the TOU Map Type must match the installation time zone.

If the Multiple Time Zone Support option is configured as "Yes" then the Time of Use Map's time zone can be set to any time zone. However, the time zone can only be changed when it is not associated to any TOU Map's that have previously generated TOU Map Data. Also when a TOU Map of this TOU Map Type has a Dynamic Option associated to it, the TOU Map Type time zone can no longer be changed.

TOU Map

Validation Algorithm: Validate Time of Use Map Dynamic Options (D2-VALTMDYOP)

This algorithm supports multiple time zone implementations. If the Multiple Time Zone Support option is configured as "Yes" then the validation algorithm will ensure:

- The Dynamic Option does not have a Dynamic Option Type with a time zone that differs from the TOU Map Type's time zone
- The Dynamic Option is not be associated to other TOU Maps that have a TOU Map Time Zone that differs from the TOU Map Type's time zone
- The Dynamic TOU Map's TOU Map Type must have the same time zone as the TOU Map Type

Dynamic Option Type

Validation Algorithm: Validate Dynamic Option Type Time Zone (D2-VALDOT'TZN)

When the Multi Time Zone Support option is not configured or set to "No" then the time zone of the Dynamic Option Type must match the installation time zone.

If the Multiple Time Zone Support option is configured as "Yes" then the Dynamic Option Type's time zone can be set to any time zone. However, the time zone must match the time zone of any TOU Map related to Dynamic Options of this type.

Calculating TOU Map data

When calculating TOU map data, the "Create TOU Map Data" algorithm (D2-CRETMD-CT) performs the following time zone conversions:

- Before deriving the TOU periods for the map data, the algorithm converts the Start Date/Time (also referred to as the Process Date/Time) from the installation time zone in standard time format to the legal time format in the time zone of the TOU Map Type (TOU Map Templates store date/times in legal format for the TOU Map Type time zone).
- After deriving the TOU periods for the map data, the algorithm converts the date/time of the first TOU period from legal time format to standard time format (because date/times for TOU Map data is stored in standard format).

Scalar Periodic Estimation

This section outlines the impact of the multiple time zone functionality on the scalar periodic estimation process.

- The First Daily Measurement Time value for a measuring component type is not anchored in a single time zone, and is interpreted as being in the local time zone of the measuring component. For example, a First Daily Measurement Time value of 12:00:00AM would be interpreted as 4/1/2015 12AM US/Pacific for a San Francisco meter and as 4/1/2015 12AM US/Eastern for a New York meter.

Because measurement data is stored in standard time, the "Incoming Data Shift Flag" (on the device or device type) can impact the First Daily Measurement Time. For example, if the First Daily Measurement Time is set to 12:00AM for a device whose Incoming Shift Flag is set to "Shift," incoming measurements would be for 12:00AM for non-DST periods, and for 11:00PM for DST periods, since the meter would actually be read at a different time in standard time (even though it would always be read at 12:00AM in wall time).

Outbound Messages

This section describes outbound message business objects provided for use with the multiple time zone functionality, including:

- XSD Usage Transaction Outbound Message (D2-XSDUsageTranOutboundMsg)
- XSD Device Event Notification (D1-XSDDeviceEventNotification)

These business objects support the multiple time zone functionality in the following ways:

- It includes a Time Zone element populated by a pre-processing algorithm. The time zone is based on the local time zone of the usage subscription or device (as appropriate).

It is important to note that this Time Zone element is not the time zone in which the data will be represented in the outbound message, which will always be installation standard time. The Time Zone element is intended to be used by external systems to convert from the installation standard time to local time of the usage subscription if necessary.

- All date/time elements will be set to standard format (stdTime="true") to ensure an accurate conversion to the XSD date/time format. This is important for the Fall DST shift day when there is a duplicate hour.
- Date/times are in the installation time zone, appended with an appropriate UTC offset, based on the time zone element (Example: 2015-05-04T11:35:55Z+5).
- This business object does not contain elements for "local" vs "standard" date/times. This business object should only be used for XSD-based requests, which do not require multiple date/time fields since the XSD format provides sufficient information to spawn a local or standard version of any date/time in the receiving application.

The "XSD Usage Transaction Outbound Message" business object also contains logic to handle converting from a wall time to a standard time to disambiguate the duplicate hours that appear on the day of the "fall back" DST transition.

Aggregations

This section outlines how the aggregation functionality supports the use of multiple time zones.

Overview

In aggregation there are two use cases for how data should be aggregated for service points that exist in differing time zones:

- **Absolute Time:** If a time zone is provided on the measuring component or measuring component type, then the aggregation horizon date/times will be used across service points of differing time zones. For example the 4/1/2015 12AM interval in US/Pacific would be combined with the 4/1/2015 3AM interval in US/Eastern. Since all date/times are stored in the installation time zone there is no need to do time zone conversions.
- **Local Time:** If no time zone is specified, consumption is aggregated by converting the aggregation horizon to the appropriate time zone for each service point. For example the 4/1/2015 12AM interval in US/Pacific would be combined with the 4/1/2015 12AM interval in US/Eastern.

Aggregation Measuring Components in general do not have any time zone related dependencies with other master configuration objects. However, when an aggregator is listed as a direct channel measuring component on a usage subscription it share the same time zone as the Usage Subscription.

Aggregation-Related Algorithms

The following algorithms used by the aggregation functionality behave differently when the multiple time zone feature is enabled.

Algorithm	Algorithm Code
Aggregate Measurements of Aggregator's Constituent MCs	D2-AGG-MC
Find Constituent Items Based on Postal and Service Type	D2-DETCITMPS
Aggregate Item Consumptions to Produce Max or Sum	D2-AGGITCMS
Find Constituent Measuring Components	D2-FIND-CMC
Aggregate Measurement Counts and Quantity	D2-AGG-MCQ

Refer to the Detailed Description on the Algorithm Type for more detailed information about these algorithms.

Smart Grid Gateway Adapters

This section outlines how the Oracle Utilities Smart Grid Gateway adapters support the multiple time zone functionality.

Landis+Gyr

The Landis+Gyr adapter supports multiple time zones for initial measurement and device event upload.

Initial Measurement Data Upload - Date/Time Format Conversion

The Landis+Gyr date/time format (MMDDYYYYHHMMSSAM) is converted to the XSD format (YYYYMMDDTHH:MM:SS). For example, if the incoming date was 11212014020000AM, the resulting date/time would be: 2014-11-21T02:00:00.

The "dateTimeInUTC" parameter in the EnvironmentSettings.xq file can be used to indicate whether Landis+Gyr is sending date/time information in UTC or local time of the device. If not provided the default behavior will be local time of the device.

If the "dateTimeInUTC" environment variable indicates that Landis+Gyr is operating in UTC then the converted date/time will have "Z" appended to the end. For example, if the incoming date was 11212014020000AM the resulting date/time would be: 2014-11-21T02:00:00Z.

Event Data Upload - Date Time Format Conversion

The XSD date/time that is received from Landis+Gyr is passed to the device event seeder with no change.

Echelon

The Echelon adapter supports multiple time zones for initial measurement and device event upload. All Echelon date/times have "Z" appended to the end because Echelon sends dates in UTC.

Initial Measurement Data Upload - Date Time Format Conversion

The Echelon date/time format (YYYY-MM-DD HH:MM:SSS) is converted to the XSD format (YYYYMMDDTHH:MM:SS). For example, if the incoming date was 2014-11-21 02:00:000 the resulting date/time would be: 2014-11-21T02:00:00Z.

Event Data Upload - Date Time Format Conversion

The Echelon date/time format (YYYY-MM-DD HH:MM:SSS) is converted to the XSD format (YYYY-MM-DD-THH:MM:SS). For example, if the incoming date was 2014-11-21 02:00:000 the resulting date/time would be: 2014-11-21T02:00:00Z.

MV90 Adapter for Itron

The MV90 Adapter for Itron supports multiple time zones for initial measurement upload.

Initial Measurement Data Upload - Date Time Format Conversion

The MV90 date/time format (MMDDYYhhmmss) is converted to XSD format (YYYY-MM-DD-THH:MM:SS). For example, if the incoming date was 112114020000 the resulting date/time would be: 2014-11-21T02:00:00.

Sensus

The Sensus adapter supports multiple time zones for initial measurement and device event upload.

Initial Measurement Data Upload - Date Time Format Conversion

The SENSUS date/time format (YYYYMMDDHHMM) is converted to the XSD format (YYYY-MM-DDTHH:MM:SS). For example, if the incoming date was 201411210200 the resulting date/time would be: 2014-11-21T02:00:00.

The "dateTimeInUTC" parameter in the EnvironmentSettings.xq file can be used to indicate whether Sensus is sending date/time information in UTC or local time of the device. If not provided the default behavior will be local time of the device.

If the "dateTimeInUTC" environment variable indicates that SENSUS is operating in UTC then the converted date/time should have "Z" appended to the end. For example, if the incoming date was 201411210200 the resulting date/time would be: 2014-11-21T02:00:00Z.

Event Data Upload - Date Time Format Conversion

The SENSUS date/time format (YYYYMMDDHHMM) is converted to the XSD format (YYYY-MM-DDTHH:MM:SS). For example, if the incoming date was 201411210200 the resulting date/time would be: 2014-11-21T02:00:00.

The "dateTimeInUTC" parameter in the EnvironmentSettings.xq file can be used to indicate whether Sensus is sending date/time information in UTC or local time of the device. If not provided the default behavior will be local time of the device.

If the "dateTimeInUTC" environment variable indicates that SENSUS is operating in UTC then the converted date/time should have "Z" appended to the end. For example, if the incoming date was 201411210200 the resulting date/time would be: 2014-11-21T02:00:00Z.

Online Creation of Initial Measurements

When creating initial measurements online from the 360 Degree View portals, initial measurements are created as appropriate for measuring components with time zones other than the installation time zone.

For example, if you copy measurements from a measuring component in one time zone to a measuring component in a different time zone, the measurements are being moved in absolute time (i.e. the 3:00 PM interval in Pacific time would be changed 6:00PM in Eastern time).

Implementing Multiple Time Zones

This section provides a high-level overview of the steps involved in implementing multiple time zones within Oracle Utilities Meter Data Management or Oracle Utilities Smart Grid Gateway. These include:

- **Define Time Zones**
- **Enable Multiple Time Zone Functionality**
- **Create Service Providers**
- **Create Master Data**
- **Create TOU Data**
- **Configure Data Synchronization**
- **Configure IMD Seeder and Device Event Seeder**
- **Configure Outbound Messages for Usage Transactions and Device Events**
- **Configure SGG Adapters**

Define Time Zones

Before you can use the multiple time zone feature, you must first define the specific time zones that will be used by your implementation. The base package provides a number of pre-defined time zones. If the time zones you plan to use are not included in the base package, you can define new time zones using the **Time Zone** portal.

Refer to the **Defining Time Zones** section in the *Oracle Utilities Application Framework Administration Guide* for more information about defining time zones.

Enable Multiple Time Zone Functionality

Enable the Multiple Time Zone functionality by setting the "Multi Time Zone Support" option to "Yes" (D1YS).

Refer to the **Defining Feature Configurations** section in the *Oracle Utilities Application Framework Administration Guide* for more information about defining feature configuration options.

Create Service Providers

Create a service provider for each external system that will either send information to or receive information from your implementation. This can include head-end systems as well as external applications such as a customer information system like Oracle Utilities Customer Care and Billing.

For each service provider, specify how date/times are sent as appropriate using the fields outlined in the following table:

Service Provider Type	Field
Head-End System (D1-HeadEndSystem)	IMD Import Date/Time Format
	Event Data Import Date/Time Format
External Application (D1-ExternalApplication)	Usage Transaction Import Date/Time Format

Create Master Data

When creating devices, measuring components, and other master data, specify the appropriate time zone for each:

- Device Configuration
- Service Point
- Usage Subscription

Remember that the time zone for related objects must match - the time zone for a device configuration must match the time for the service point where it's installed, and the service point's usage subscription(s).

Create TOU Data

If your implementation uses time of use calculations, indicate the appropriate time zone for each TOU Map Type and Dynamic Option Type.

Remember that the time zones for TOU Map Types and related Dynamic Option Types must match.

Configure Data Synchronization

If your implementation will use data synchronization between systems, be sure to configure the "Validate and Default Sync Request Inbound Time Zone" algorithm (D1-VALDFSITZ) as appropriate. The "Error if Time Zone Not Populated" algorithm parameter can be used to require that the time zone must be populated on incoming synchronization requests.

Configure IMD Seeder and Device Event Seeder

The IMD Seeder and Device Event Seeder business objects must be configured to validate time zone information for incoming initial measurements and device events.

IMD Seeder

Use the "Error if MC is Not Installed" parameter on the "Derive Service Provider and Measuring Component" algorithm (D1-DER-SPRMC) to prevent initial measurements from being processed for devices that are not installed where the incoming initial measurement does not have any time zone indications.

Note: This is only necessary if you anticipate that you will receive device configuration synchronizations with the wrong time zone and only correct them once the service point is synchronized. If device configuration synchronizations have the correct time zones, this option is not needed.

Device Event Seeder

Use the "Error if Device is Not Installed" parameter on the "Shift Event Date/Times to Standard" algorithm (D1-SHEVTDTTM) to prevent device events from being processed for devices that are not installed where the incoming device event does not have any time zone indications.

Configure Outbound Messages for Usage Transactions and Device Events

If your implementation will send usage transactions and/or device events to subscribing systems, you must create outbound message types that reference the following business objects:

- XSD Usage Transaction Outbound Message (D2-XSDUsageTranOutboundMsg)
- XSD Device Event Notification (D1-XSDDeviceEventNotification)

Configure SGG Adapters

If your implementation includes Oracle Utilities Smart Grid Gateway and uses either the Landis+Gyr or Sensus adapters, you must configure the `EnvironmentSettings.xq` files to indicate whether the adapters are sending date/time information in UTC or local time of the device. If not provided the default behavior will be local time of the device.

