Oracle Utilities Smart Grid Gateway

Configuration Guide

Release 2.1.0 Service Pack 3

E41191-03

May 2015

ORACLE®

Oracle Utilities Smart Grid Gateway/Smart Grid Gateway Installation and Configuration Guide, Release 2.1.0 Service Pack 3

E41191-03

Primary Author: Lou Prosperi

Contents

Chapter 4

Chapter 5

Appendix A

Appendix B

Appendix C

Appendix D

What's New

New Features in the Oracle Utilities SMart Grid Gateway 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...
Service Order Management	A separate configuration guide is provided for Service Order Management.	**Oracle Utilities Smart Grid Gateway Service Order Management Configuration Guide**
Information Lifecycle Management	Added an appendix describing how Information Lifecycle Management (ILM) works with Oracle Utilities Smart Grid Gateway.	**Appendix C**: **Information Lifecycle Management**
Multiple Time Zone Support	Added an appendix describing how multiple time zones are supported in Oracle Utilities Smart Grid Gateway.	**Appendix D**: **Multiple Time Zone Support**

Chapter 1

Overview

This chapter provides an overview of this configuration guide and an introduction to the Oracle Utilities Smart Grid Gateway application. This includes:

- **What Is This Book?**
- **Architecture Overview**
- **Oracle Utilities Application Framework Configuration Tools**
- **Configuration Process Overview**

What Is This Book?

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

Oracle Utilities Smart Grid Gateway 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 Smart Grid Gateway.

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** on page 3-7 in **Chapter 3**: **Setting Up Admin Data** 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 Smart Grid Gateway 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 Smart Grid Gateway.
- **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**: **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 8**: **Batch Processing** provides a list of the base package batch controls provided with the system.
- **Chapter 9**: **Integrating Oracle Utilities Smart Grid Gateway with Other Systems** provides a description of how Oracle Utilities Smart Gird Gateway can be integrated with other systems.
- **Chapter 10**: **Configuration Examples and Sample Implementation** provides a high-level description of a sample implementation.

- **Chapter 11**: **Smart Grid Gateway Adapters** provides a description of the components included in Oracle Utilities Smart Grid Gateway adapters.
- **Appendix A**: **Glossary** is a list of commonly used terms.
- **Appendix B**: **Configuration Migration Assistant** provides information about how to use the Configuration Migration Assistant feature of the Oracle Utilities Application Framework with Oracle Utilities Smart Grid Gateway.
- **Appendix C**: **Information Lifecycle Management** provides information about how to use the Information Lifecycle Management feature of Oracle Database with Oracle Utilities Smart Grid Gateway.
- **Appendix D**: **Multiple Time Zone Support** provides information about support for multiple time zones in Oracle Utilities Smart Grid Gateway.

Other Documentation

This section describes other documentation provided with Oracle Utilities Smart Grid Gateway.

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 Smart Grid Gateway Quick Install Guide
- Oracle Utilities Smart Grid Gateway DBA Guide
- Oracle Utilities Smart Grid Gateway 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 Smart Grid Gateway Adapter for Landis+Gyr User's Guide
- Oracle Utilities Smart Grid Gateway MV90 Adapter for Itron User's Guide
- Oracle Utilities Smart Grid Gateway Adapter for Echelon User's Guide
- Oracle Utilities Smart Grid Gateway Adapter for Sensus User's Guide
- Oracle Utilities Smart Grid Gateway Adapter for Silver Springs Network User's Guide
- Oracle Utilities Smart Grid Gateway Adapter Development Kit User's Guide
- Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay User's Guide

Adapter Configuration Documentation

Adapter configuration documentation provides technical information related to vendor-specific Smart Grid Gateway adapters, and includes the following documents:

- Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr Configuration Guide
- Oracle Utilities Smart Grid Gateway MV90 Adapter for Itron Configuration Guide
- Oracle Utilities Smart Grid Gateway Adapter for Echelon Configuration Guide
- Oracle Utilities Smart Grid Gateway Adapter for Sensus Configuration Guide

- Oracle Utilities Smart Grid Gateway Adapter for Silver Springs Network Configuration Guide
- Oracle Utilities Smart Grid Gateway Adapter Development Kit Configuration Guide
- Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay Configuration Guide

Service Order Management Documentation

Service order management documentation provides technical information related to Service Order Management, and includes the following documents:

- Oracle Utilities Smart Grid Gateway Service Order Management Configuration Guide
- Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Service Order Management Implementation Guide
- Oracle Utilities Service Order Management Integration to Oracle Utilities Mobile Workforce Management Implementation Guide

Supplemental Documentation

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

- Oracle Utilities Smart Grid Gateway Server Administration Guide
- Oracle Utilities Smart Grid Gateway Batch Server Administration Guide

Oracle Utilities Smart Grid Gateway adapters use Oracle Service Bus (OSB) and Oracle Business Process Execution Language (BPEL) as middleware components. These tools are part of the Oracle SOA Suite. See the Oracle SOA Suite Documentation library (http://www.oracle.com/technetwork/middleware/soasuite/documentation/index.html) for more information about using these tools.

Embedded Help

Oracle Utilities Smart Grid Gateway, 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 Smart Grid Gateway also include 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 Smart Grid Gateway Adapters: Describes the features and functions provided in the smart grid gateway adapters (D3, D4, D5, etc.)

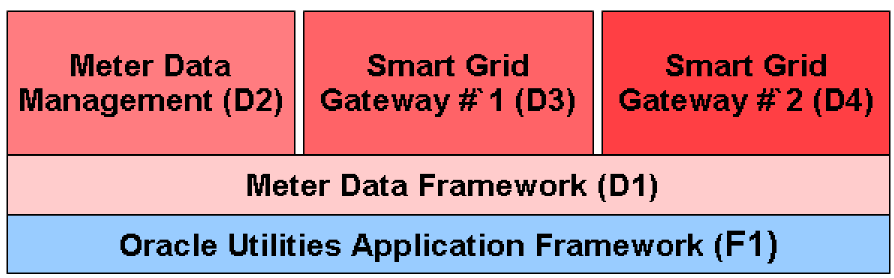
Architecture Overview

Oracle Utilities Smart Grid Gateway 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 issue commands to meters via head-end systems.

Oracle Utilities Smart Grid Gateway 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 provide consumer-specific processing of meter events and activities
- Device Communication: the maintenance of communications with head-end systems, including import of usage and events, as well as two-way communications used in issuing meter commands.
- Validation, Editing, and Estimation: the maintenance of measurement data and the engine used to validate and modify that data as it comes in

Oracle Utilities Smart Grid Gateway 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 Smart Grid Gateway 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 Smart Grid Gateway 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 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 data areas, and Oracle recommends that you take advantage of data areas where possible to avoid redundant data definition.

See **Appendix B**: **Base Package Data Areas** for a list of the base package data areas provided with Oracle Utilities Service and Measurement Data Foundation and Oracle Utilities Smart Grid Gateway.

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 a means 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 Framework documentation. To view information about specific algorithms provided with the base system, use the Application Viewer (also described in the

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 Smart Grid Gateway 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 Smart Grid Gateway, 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 Smart Grid Gateway adapter entities start with "D3", "D4", "D5", etc.

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 Smart Grid Gateway migration requests that can be used when migrating administrative data, refer to **Appendix B: 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**

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 leads 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 use for multiple measuring components.

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. 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.
7. 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 Smart Grid Gateway.

> **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 Smart Grid Gateway is centered around the creation of business objects. Nearly every object or set of data used by Oracle Utilities Smart Grid Gateway 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 Smart Grid Gateway 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)
   - 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 Smart Grid Gateway 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 Smart Grid Gateway 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 Smart Grid Gateway 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
Communication	Communication Type
Contact	Contact Type
Device	Device Type
Device Event	Device Event Type
Device Configuration	Device Configuration Type
Measuring Component	Measuring Component Type
Service Point	Service Point Type

Chapter 2

General Configuration

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

- **Installation Options**
- **Master Configuration**

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.

Master Configuration

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.

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 Smart Grid Gateway, 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 related products, including Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway.

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.

Admin Setup Reference Tables

This section lists and describes all objects that must be defined as part of the setup process for Oracle Utilities Smart Grid Gateway. 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 Smart Grid Gateway. 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 Smart Grid Gateway 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 a 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 Type	Common	Defines specific types of service for which usage can be recorded and captured (electric, gas, steam, etc.)	None
L2	Manufacturer	Device	Individual companies that makes devices. Manufacturers also reference models.	Service Type
L2	Unit of Measure	Common	Quantities measured and recorded by the system (CCF, KWH, KW, etc.)	Service 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
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
L5	Service Point Type	Device Installation	Defines specific types of points at which service is delivered	Service Type, Device Type

* 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**

Note: References to objects and options pre-fixed with “D2” are available only with Oracle Utilities Meter Data Management, and included for illustrative purposes only.

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 in 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:

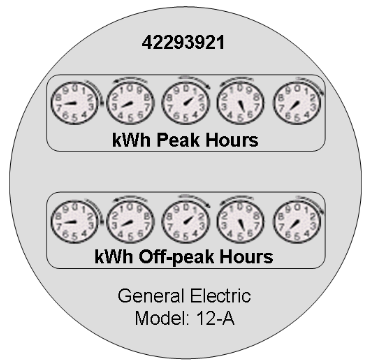
- **Unit of Measure**: The unit of measure for the quantity being recorded. Examples include kilo-watt hours (kWh), kilo-watts (kW), therms, cubic feet (CCF), temperature (Fahrenheit or Celsius), etc.
- **Time of Use**: Modifiers for a given unit of measure that indicate a period of time during which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

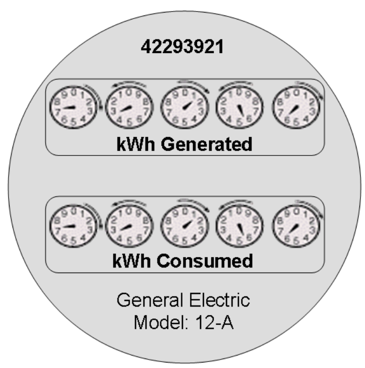
- **Service Quantity Identifiers**: Used to further distinguish between measured quantities that have identical UOM/TOU combinations, including situations in which the distinguishing identifier of a UOM is not accurately described as a TOU. Generally, SQI is only used when multiple measuring components 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:

Usage = End Measurement - 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 an 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 import 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	• 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 additionl 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	• 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:
	  • 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	• Information: D1-SMARTINFO (Smart Meter Information) • Validation: D1-VALRETDT (Validate Device Retirement Date)
Lifecycle	• 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	• 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 additionl 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	• 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 - Retreive Details for Display) • Maintenance UI Map: D1-DeviceConfigMaint (Device Configuration- Maintenance)

Option	Description
Algorithms	• 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-INSTDCVAL (Earlier Installed Device Configuration) • Validation: D1-VALTIMZON (Validates BO Time Zone value against Installation Option)
Lifecycle	• 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	• 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	Activity Statistics Aggregator Type Instances of this business object represent individual activity statistics aggregator measuring component types defined in the system. Activity Statistics Aggregators are used to track statistics related to device event and measurement uploads.
D1-AutoReadRegisterType	Auto-Read Register Type Instances of this business object represent individual auto-read register measuring component types defined in the system. 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.
D1-GenericMCType	Generic MC Type (used as a parent measuring component type)
D1-IntervalChannelTypePhysical	Interval Channel Type - Physical Instances of this business object represent individual physical interval channel measuring component types defined in the system.
D1-IntervalChannelTypeScratchp	Interval Channel Type - Scratchpad Instances of this business object represent individual scratchpad interval channel measuring component types defined in the system.
D1-RegisterTypePhysical	Register Type Instances of this business object represent individual physical register measuring component types defined in the system.

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	• 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	• 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.

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	• 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	• 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:
	  • 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:
	  • 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	• 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.

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**

Note: References to objects and options pre-fixed with “D2” are available only with Oracle Utilities Meter Data Management, and included for illustrative purposes only.

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 a 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	• 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	• 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	• 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) * Available with Oracle Utilities Meter Data Management.
Lifecycle	• Active (Initial) • Inactive (Final)

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

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	• 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	• Portal Navigation Option: d1fcltyTabMenu (Facility Maintenance)
Algorithms	• Information: D1-TRNSFINFO (Transformer Information) • Validation: D1-FCLTSTVAL (Facility Service Type Validation)
Lifecycle	• 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	• 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	Network Location Instances of this business object represent individual network locations defined in the system.

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	• Portal Navigation Option: d1nwlocTabMenu (Network Location)
Algorithms	• Information: D1-NWLOCINFO (Electricity Network Location Information) • Validation: D1-FCLTYVAL (Facility Status Validation) • Validation: D1-NWLDTVAL (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	• 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-ContactODMBORead	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	• Synchronization Add BO: D1-SynchronizationAddContact (Contact Synchronization Add) • Portal Navigation Option: d1contctTabMenu (Contact)
Algorithms	• 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	• 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	• 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	• 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	• 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	• 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	• 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-PROCMETHD 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-PROCMETHD

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

Option/Field	Description
Maintenance Object	D1-PROCMETHD
Description	Processing Method
Service Name	D1-PROCMETHOD (Processing Method Maintenance)
Tables	• 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	Cancelation 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-OMTDFTT / D1-OMTDFTT	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-PROCMETHD (Processing Method)
Parent Business Object	D1-AbstractProcessingMethod
Lifecycle Business Object	Generic Processing Method
Application Service	D1-PROCMETHD (Processing Method MO)
Instance Control	Allow New Instances
Options	• Applicable Processing Role: D1IM • Portal Navigation Option: d1svcproTabMenu (Service Provider) • Maintenance UI Map: D1-HowToCreateMCInfoMaint (How To Crt MC Related Info - Maintenance)
Algorithms	• 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.

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**
- **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 finals 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 to 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*20) and 25 (20 + 0.25*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-IntialLoadIMDScalar) 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 • 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 s 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 pass 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 SQI: n/a	Loss Adjusted UOM: CCF SQI: n/a	Thermal Units UOM: BTU SQI: 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 SQI: n/a	Loss Adjusted UOM: CCF SQI: n/a	Thermal Units UOM: BTU SQI: n/a	Normal CCF UOM: CCF SQI: Normal	Percent (%) of Normal UOM: CCF SQI: 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 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 because there 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:

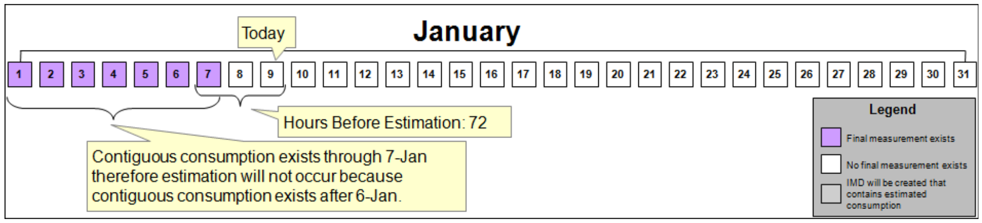
- **Hours Before Estimation**: the number of hours after the End Date and Time of the most recent measurement that must pass before the measuring component is considered due for estimation. For example, if set to 72 hours, estimation will only take place 72 hours after the End Date and Time of the latest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

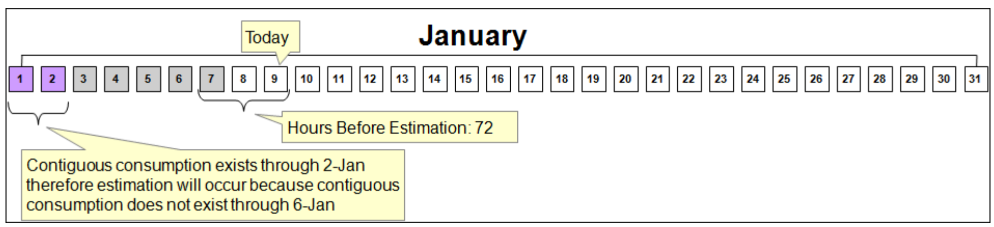
If the measuring component does NOT have contiguous final measurements on/after a date and time equal to the current date/time minus the “Hours Before Estimate” value, the measuring component is estimated. If the measuring component is subject to estimation, measurements will be estimated through the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

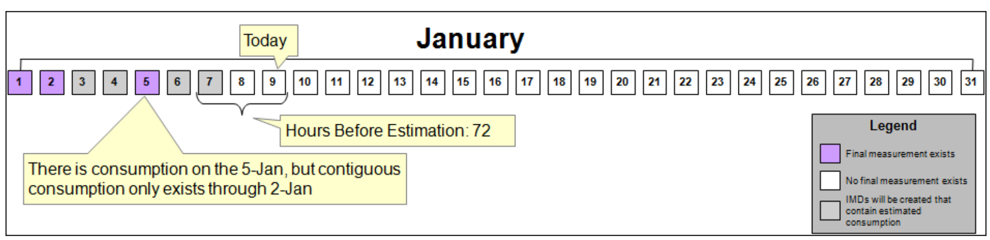
Example 1: On January 9, if there is contiguous consumption through January 7, estimation would not occur because consumption exists after January 6 (3 days prior to January 9). The diagram below illustrates this example:



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).

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	• D1_INIT_MSRMT_DATA (Initial Measurement) - Primary • D1_INIT_MSRMT_DATA_CHAR (Initial Measurement Characteristics) - Child • D1_INIT_MSRMT_DATA_LOG (Initial Measurement Log) - Child • D1_INIT_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	Intial 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-IntialLoadIMDMain	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	• 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	• 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	• 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.

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	• 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	• 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	• Audit: D1-AMSRMTLOG (Formats the standard description of a Measurement) • Information: D1-MSRMTINFO (Measurement Information)
Lifecycle	• OK (Initial) • Re-Derive Values (Interim)

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

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	• 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	Average Daily Estimated Item Consumption Instances of this business object represent individual device type service quantities defined in the system.

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	• 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

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 Commands and Device Communication**
- **Understanding Bulk Commands**
- **Understanding Device Events**
- **Activities In Detail**
- **Inbound Communications In Detail**
- **Outbound Communications In Detail**
- **Completion Events In Detail**
- **Device Events In Detail**
- **Configuring Activities and Device Events**
- **Configuring Device Communication and Device Event Objects**

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 Commands and Device Communication

This section provides an overview of entities related to device communications, including commands activities, communications, and completion events and how they are used in the Service and Measurement Data Foundation and related products, including Oracle Utilities Smart Grid Gateway.

Activities

Activities are used in Oracle Utilities Smart Grid Gateway to record meter commands such as device commissioning/decommissioning, remote connect/disconnect, device status check, or on-demand reading commands. Activities are also used in Oracle Utilities Smart Grid Gateway to track statistics related to uploading initial measurements and device events sent from head-end systems.

Commands

Commands issued to devices, such as remote connect, remote disconnect, and others are defined as activities. These represent commands sent via Oracle Utilities Smart Grid Gateway to devices to invoke a specific type of action.

The table below outlines the commands supported by Oracle Utilities Smart Grid Gateway.

Command	Description
Commission Device	A command issued to establish communication between a device and the head-end system. The goal is to ensure connectivity has been established with the device, that any information needed to communicate with the device has been defined in both Oracle Utilities Smart Grid Gateway and the head-end system, and that the device will begin capturing usage and events.
Decommission Device	A command issued to inform the head-end system when a device needs to be removed from a service point, so that no further reads or events will arrive from the device. Decommissioning is invoked when a device must be removed or deactivated. The goal is to stop any communication between the device and the head-end system.
Device Status Check	A command used to test whether the device is communicating with the network, determine the connection status of the device, and when possible, and check if there are any known malfunctions.
On-Demand Read	A request for the most up-to-date reading from a particular device. These commands are not guaranteed to return immediately. In some cases, completing the command might require a person to manually read the device. The purposes are to check the operational status of the device and/or obtain a more recent reading than is currently available.
Remote Connect	A command issued when a device needs to be connected at a service point.
Remote Disconnect	A command issued when a device needs to be disconnected or shut off at a service point.

Attributes used to define commands include the following:

- **Parent Activity:** the parent activity (if any) for the command.
- **Command Effective Date/Time**: the date and time on which the command takes effect. Commands issued prior to this date and time remain in the "Waiting for Effective Date" status until this time, at which time the command is executed.
- **Command Expiration Date/Time**: the date time when the command expires. The command cannot be executed after this date and time.
- **Priority**: the priority for the command.
- **Requester**: the application sending the command.
- **Requester User**: the user who initiated the command.
- **Requester Transaction ID**: an ID for the command, defined by the requester.
- **Utility Device ID**: ID of the device used by the utility. Used to derive the device ID if the device ID is not provided.

When a command is initiated, it in turn creates an outbound communication which sends the command request to the head-end system.

Upload Statistics

Upload statistics are statistics related to the uploading of initial measurement data and device events sent from a head-end system, and are defined as activities in the system. This section describes upload statistics and how they are captured and maintained in Oracle Utilities Smart Grid Gateway, including:

- **Upload Statistics Activities**
- **Upload Statistics - XAI Inbound Services**
- **Head-End System Processing Statistics**

Upload Statistics Activities

There are three types of upload statistics activities:

- **Payload Statistics**: Contains statistics related to a specific payload (file) containing one or more initial measurements or device events. Payload Statistics activities contain:
    - Basic information about the payload (head-end system, file name, and status)
    - Middleware statistics including specifics about the file, the total number of initial measurements or device events processed, the number of initial measurement or device events errors, and total processing time
    - Initial measurement statistics including the number of initial measurements processed
    - Device event statistics including the number of device events processed
- **Payload Error Notification**: Contains details concerning processing errors encountered in an individual payload (file) containing one or more initial measurements or device events. Payload Error Notification activities are related to Payload Statistics activities.
- **Payload Summary**: Contains processing summary statistics for an individual payload (file) containing one or more initial measurements or device events. Payload Summary activities are related to Payload Statistics activities, and are used to update related payload statistics upon the completion of payload processing.

Upload statistics activities are created during processing of payload files as follows:

- When processing begins for a payload, a Payload Statistics activity is created to record the process.

- If an error occurs during processing, a Payload Error Notification activity is created.
- When payload processing is complete, a Payload Summary activity is created, which in turn, updates the Payload Statistics activity with details concerning the processing of the payload, including the start and end time of the processing, the total processing time, the number of initial measurements or device events processed, and the number of initial measurement or device event errors (if any).

Upload Statistics - XAI Inbound Services

Upload statistics activities are created by the middleware components used by Smart Grid Gateway adapters via XAI inbound services. XAI inbound services define the details of how messages are received from an external system, including the activity business object to be invoked when the response message is received. Smart Grid Gateway adapters use a set of XAI inbound services to create upload statistics activities, outlined in the table below:

This type of Upload Statistics Activity...	Is created by this type of XAI Inbound Service
D1-PayloadStatistics	D1-PayloadStatistics
D1-PayloadSummary	D1-PayloadSummary
D1-PayloadErrorNotif	D1-PayloadErrorNotif

Refer to the Oracle Utilities Application Framework documentation for more information about XAI Inbound Services.

Head-End System Processing Statistics

As Oracle Utilities Smart Grid Gateway processes payloads containing initial measurements or device events, statistics for each payload are captured in Payload Statistics activities. Over time, payload statistics for each head-end system are summarized to allow administrators to view summary statistics for the head-end system. These summarized statistics are referred to as head-end system processing statistics.

Head-end system processing statistics are stored as aggregated measurements for aggregator measuring components. A separate aggregator measuring component must be set up for each head-end system for which processing statistics will be aggregated.

Communications

Communications are records of messages sent between Oracle Utilities Smart Grid Gateway and an external system, such as a head-end system or edge application as a result of initiating a command for a device. Communications can flow both outbound and inbound. Communications are most often created as a result of a command activity.

Attributes used to define communications include the following:

- **Device ID**: the ID of the device related to the communication. All communications (and their related commands) are related to a device.
- **AMI Device Identifier Number**: the identifier for the device used by the head-end system.
- **Event Date/Time**: the date and time of the message.
- **Command Information**: details concerning the command that created the communication, including:
  - **Recipient**: the recipient of the command (recipients are defined as service providers)
  - **Transaction ID**: an ID for the command that created the communication.
  - **External Transaction ID**: ID for the command that created the communication in the external system that sent or received the communication.

- **Event Date/Time**: the date and time of the command that created the communication.

See **Understanding the Command Communication Process** below for more information about the role of communications in the device communication process.

Outbound Communications

Outbound Communications represent messages sent from Oracle Utilities Smart Grid Gateway to an external system, such as a head-end system or edge application (such as Oracle Customer Care and Billing). Outbound communications use the following types of objects:

- **Outbound Communication Business Objects**
- **Outbound Message Types**
- **XAI Senders**
- **External Systems**

Outbound Communication Business Objects

An outbound communication business object must be created for each type of message to be sent to an external system. For head-end systems, this is based on the types of messages the system is designed to accept. For example, suppose a head-end system supports the types of commands outlined above (device commission, device decommission, device status check, on-demand readings, remote connect, and remote disconnect), and that this head-end system accepts a separate type of message for each command. For this example, you would need to create outbound communication business objects for each command, as follows:

Command	Outbound Communication Business Object
Commission Device	Commission Device Outbound Communication
Decommission Device	Decommission Device Outbound Communication
Device Status Check	Device Status Check Outbound Communication
On-Demand Read	On-Demand Read Outbound Communication
Remote Connect	Remote Connect Outbound Communication
Remote Disconnect	Remote Disconnect Outbound Communication

Outbound Message Types

A outbound message type must also be created for each type of message to be sent to an external system. Again, this is based on the types of messages the system is designed to accept. To continue the example above, you might create the following outbound message types:

Command	Outbound Message Type
Commission Device	Commission Device
Decommission Device	Decommission Device
Device Status Check	Device Status Check
On-Demand Read	On-Demand Read
Remote Connect	Connect Device
Remote Disconnect	Disconnect Device

Refer to the Oracle Utilities Application Framework documentation for more information about outbound message types.

XAI Senders

You must also create an XAI Sender for each type of message to be sent to an external system. XAI senders define the details of how messages are sent to an external system. As in the case of outbound communication business objects and outbound message types, the set of XAI senders you need to create is based on the types of messages the system is designed to accept. To continue the example above, you might create the following XAI senders:

Command	XAI Sender
Commission Device	Commission Device
Decommission Device	Decommission Device
Device Status Check	Device Status Check
On-Demand Read	On-Demand Read
Remote Connect	Connect Device
Remote Disconnect	Disconnect Device

Refer to the Oracle Utilities Application Framework documentation for more information about XAI senders.

External Systems

You must also create an External System for each external system to which Oracle Utilities Smart Grid Gateway will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch, XAI, or Real-time)
- The corresponding XAI senders
- Batch Control (if Processing Method is set to Batch)
- Message XSL, W3C Schema, and Response XSL (as applicable)

To continue the example above, you might create the following external system:

Head-End System			
Outbound Message Type	**Processing Method**	**XAI Sender**	**Message XSL / Response XSL**
Commission Device	Real-time	Commission Device	HES-Request.xsl / HES-Response.xsl
Decommission Device	Real-time	Decommission Device	HES-Request.xsl / HES-Response.xsl
Device Status Check	Real-time	Device Status Check	HES-Request.xsl / HES-Response.xsl
On-Demand Read	Real-time	On-Demand Read	HES-Request.xsl / HES-Response.xsl
Remote Connect	Real-time	Connect Device	HES-Request.xsl / HES-Response.xsl
Remote Disconnect	Real-time	Disconnect Device	HES-Request.xsl / HES-Response.xsl

Refer to the Oracle Utilities Application Framework documentation for more information about external systems.

Inbound Communications

Inbound Communications represent messages sent from a head-end system or edge application (such as Oracle Customer Care and Billing) to Oracle Utilities Smart Grid Gateway. Inbound communications are typically sent in response to a command. Inbound communications use the following types of objects:

- **Inbound Communication Business Objects**
- **XAI Inbound Service**

Inbound Communication Business Objects

An inbound communication business object must be created for each type of message to be received from an external system. For head-end systems, this is based on the types of messages the system is designed to send. To continue the above example, a head-end system supports the types of commands outlined above (device commission, device decommission, device status check, on-demand readings, remote connect, and remote disconnect), and sends a separate type of message in response to each command. For this example, you would need to create the following inbound communication business objects:

Command Being Responded To	Inbound Communication Business Object
Commission Device	Commission Device Response
Decommission Device	Decommission Device Response
Device Status Check	Device Status Check Response
On-Demand Read	On-Demand Read Response
Remote Connect	Remote Connect Response
Remote Disconnect	Remote Disconnect Response

XAI Inbound Service

You must also create an XAI Inbound Service for each type of message to be received from an external system. XAI inbound services define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound communication business objects, the set of XAI inbound services you need to create is based on the types of messages the system is designed to send. To continue the example above, you might create the following XAI inbound services:

XAI Inbound Service	Schema (Inbound Communication Business Object)
Commission Device Response	Commission Device Response
Decommission Device Response	Decommission Device Response
Device Status Check Response	Device Status Check Response
On-Demand Read Response	On-Demand Read Response
Connect Device Response	Remote Connect Response
Disconnect Device Response	Remote Disconnect Response

Refer to the Oracle Utilities Application Framework documentation for more information about XAI Inbound Services.

Completion Events

Completion events are used to create or update data to reflect the effect of an activity or command. Completion events are created upon successful receipt of inbound communications related to an activity or command. For example, a commission device command could result in the creation or update of an install event, while a on-demand read command could result in the creation of an initial measurement.

Attributes used to define completion device events include the following:

- **Activity**: the activity (command) that initiated the completion event.
- **Sequence**: defines the relative order by which completion events for the activity are executed (in the event that more than one completion event is created for an activity).
- **Inbound Communication**: the inbound communication that triggered the completion event.
- **Event Date/Time**: the date and time of the completion event.

Several of the commands supported by Oracle Utilities Smart Grid Gateway have related completion events. The table below describes the completion events for each command.

Command	Completion Event
Commission Device	Device Commissioning Completion Event • Creates an install event for the device (if one doesn't exist) • Updates the device's install event status to reflect that the device has been commissioned
Decommission Device	Device Decommissioning Completion Event • Updates the device's install event status to reflect that the device has been decommissioned
On-Demand Read	Create IMD Completion Event • Creates an initial measurement for the device
Remote Connect	Connect Device Completion Event • Updates the device's install event status to reflect that the device has been connected
Remote Disconnect	Disconnect Device • Updates the device's install event status to reflect that the device has been disconnected

See **Understanding the Command Communication Process** below for more information about the role of completion events in the device communication process.

Understanding the Command Communication Process

This section provides an overview of the communication process that takes place when a command is initiated for a device. 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 Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr. Refer to the *Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr Configuration Guide* for more details about the configuration objects used by this adapter.

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

Step	Process	Landis+Gyr Adapter Objects
1.	A user initiates a remote connect command for a device.s A remote connect activity business object is instantiated for the command.	Activity BO: Remote Connect (D1-RemoteConnect)
2.	The remote connect command activity business object creates an outbound communication. The specific type of outbound communication business object created is determined by the head-end system service provider (based on the processing role defined in the Enter algorithm of the “Awaiting Response” status of the outbound communication business object’s lifecycle.	Outbound Communication BO: Initiate Connect Disconnect (D3-InitiateConnectDisconnect)
3.	The outbound communication creates an outbound message. The specific type of outbound message created is determined by the head-end system service provider. (based on the processing role defined in the Enter algorithm of the “Awaiting Response” status of the outbound communication business object’s lifecycle).	Outbound Message Type: Connect Device (D3-CONNECT)
4.	The outbound message is sent to middleware components via an External System and XAI Sender. Middleware components utilize Business Process Execution Language (BPEL).	External System: Landis+Gyr Head End System XAI Sender: Connect Device (D3-Connect)
5.	The middleware converts the outbound message from SGG format into the format used by the head-end system, and sends the message to the head-end system.	
6.	When the head-end system sends a response, the middleware receives the response message from the head-end system, and converts it from the format used by the head-end system to SGG format and invokes an XAI Inbound Service.	XAI Inbound Service: D3-ConDisconStChgNotification (D361040925)
7.	The XAI Inbound Service picks up the message, and creates a corresponding inbound communication. The specific type of inbound communication business object created is determined by the XAI Inbound Service.	XAI Inbound Service: D3-ConDisconStChgNotification (D361040925) Inbound Communication BO: Connect Disconnect State Changed Notification (Multispeak) (D3-ConnectDisconStateChgNtf)
8.	The inbound communication identifies the parent outbound communication.	Outbound Communication BO: Initiate Connect Disconnect (D3-InitiateConnectDisconnect)

Step	Process	Landis+Gyr Adapter Objects
9.	The inbound communication creates a completion event to update the status of the device to indicate it has been connected. The specific type of completion event business object created is specified in an Enter algorithm on the “Create Completion Event” Status of the inbound communication business object’s lifecycle.	Completion Event BO: Connect Device (D1-ConnectDevice) Algorithm: D3-CCE (Create Completion Event)
10.	The inbound communication updates the outbound communication. This update is performed by an Enter algorithm on the “Completed” Status of the inbound communication business object’s lifecycle.	Outbound Communication BO: Initiate Connect Disconnect (D3-InitiateConnectDisconnect)
11.	The outbound communication updates the “Connect/Disconnect Completion Flag” and the original activity business object. This update is performed by an Enter algorithm on the “Completed” Status of the outbound communication business object’s lifecycle.	Outbound Communication BO: Initiate Connect Disconnect (D3-InitiateConnectDisconnect)

Understanding Bulk Commands

A bulk command is a command that is sent to multiple devices. Examples include using a bulk command to send commissioning commands to a set of meters installed on a particular day, or to send status check commands to a set of meters to verify that power has been restored after an outage. This section describes bulk commands, including:

- **Bulk Command Processing Overview**
- **Submitting Bulk Commands Requests**

Bulk Command Processing Overview

A bulk command is initiated when an external application or an online user sends a request for a bulk command to a BPEL process. The request contains a collection of meters or devices against which a particular action needs to be taken. BPEL parses the request and sends individual command requests to the Oracle Utilities Service and Measurement Data Foundation for each meter. BPEL also creates a batch request header to track the activities associated with the bulk command. If any failure occurs in the process, a failure notification will be sent to the requesting application. Upon completion, BPEL sends a notification to the Service and Measurement Data Foundation to indicate that the process is completed.

This following table describe bulk command processing in greater detail.

Step	Process	Description
1.	A bulk request is initiated.	Bulk requests can be initiated from an external system or by way of an online request. A bulk request contains the following information: • The command to execute • The response creation method • External IDs for the bulk request and requestor • List of devices to which the command should be applied • Expiration date/time. This is the date and time after which the commands should be discarded if it has not completed successfully.
2.	A bulk request header is created.	The bulk request header is used to track the status of the individual commands that are linked to the bulk request. It also references the response creation method associated with the request.
3.	Individual commands are created.	A separate command is created for each device listed in the bulk request. Individual commands are linked to their associated bulk request header.
4.	Individual commands are initiated.	The bulk request header is updated to reflect the number of commands successfully initiated. If an individual command is not valid for a specified device, an error is generated for that command.
5.	Individual commands are executed.	Aside from their link to their associated bulk request header, these commands are identical to regular device commands.

Step	Process	Description
6.	Responses are sent to the external system.	In the case of bulk command requests submitted from an external system, as individual commands are completed (when they reach their final state), a response is sent to the external system. The specific manner in which this notification is sent is based on the specified response creation method (see **External Request for Bulk Commands** on page 7-14 for more information).

Submitting Bulk Commands Requests

You can request a bulk command either online, from Oracle Utilities Smart Grid Gateway, or externally, from another application.

External Request for Bulk Commands

When an external application sends a request for a bulk command, a BPEL process parses the bulk request and creates a set of individual commands that are sent to each device listed in the bulk request. This BPEL process also creates the bulk request header that is used to track the progress of the commands and to send one or more responses back to the external application.

The responses are sent according to the response creation method included in the bulk request. The following response creation methods are supported:

- **One-by-One:** The commands are processed individually and a response is sent as each command reaches its final state. The external bulk request ID is included on each individual command response that is sent.
- **Wait for All:** The commands are processed individually and a response is sent when all of the commands have reached their final state. The first command to reach a final state creates the bulk response, which is updated as each additional command reaches its final state.
- **Timed:** The commands are processed individually and a response is sent every x minutes. The first command to reach its final state creates a bulk response. Each additional command to reach its final state during the timed interval updates the bulk response, which is then sent at the end of the interval. This process repeats until all command activities are in their final state, or until the bulk command request expires. The number of minutes for the timed interval is defined on the bulk request header activity type.

When all individual commands are in their final state, the bulk header transitions to its final state. The Activity portal for the bulk header lists information about the bulk command, such as status reasons for discarded commands and command statistics.

Online Request for Bulk Commands

Users can also request a bulk command online from the Device Query portal. When the request is made, the application creates a bulk request header to track the progress of the individual commands sent to the devices. If an individual command is not valid for a particular device, an error is generated for that command. Each individual command is linked to the bulk request header by the activity relationship type flag Online Bulk Request Header. The Activity portal for the bulk header lists information about the bulk command, such as command statistics and status reasons for discarded commands.

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 alter 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.	The head-end system sends a payload of device events to middleware components. The payload contains both standard and paired device events. Oracle Utilities Smart Grid Gateway adapters use Oracle Service Bus. (OSB) as the middleware components for import of usage readings and device events,	
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.	Pre-Processing Algorithm: D1-SPRID (Service Provider Identification) Pre-Processing Algorithm: D1-DEVICEID (Device Identification) Pre-Processing Algorithm: D1-SHEVTDTTM (Shift Event Date/Times to Standard)
5.	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. 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.	Pre-Processing Algorithm: D1-DETBOID (Device Event Type and Business Object Identification) Processing Method: D1-HowToProcessDeviceInfo (How to Process Device Related Information) Device Event Mapping BO: D1-DeviceEventMappingLookup
6.	An instance of the appropriate device event business object is created in the system for each device event received. For paired events, the first event business objects may create activities to represent the event.	Standard Device Event BO: D1-StandardDeviceEvent Paired (First) Device Event BO: D1-PairedEventFirstDeviceEvent (Device Event - Paired Event (First) Paired (Last) Device Event BO: D1-PairedEventLastDeviceEvent (Device Event - Paired Event (Last)
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.	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. Sending device event information is triggered by an Enter algorithm on the "Processed" Status of the device event business object's lifecycle. 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.	Processing Method: D1-HowToProcDveEvtsInformation (How to Process Device Event Related Info)

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.

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	• 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-RelMCREAbtract	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-UTCorrectnPrcessrActvity	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	• 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-RtDDADtls (Device Activity with Duration - Retreive Details for Display) • Maintenance UI Map: D1-DvcWithDuratnActivityMaint (Outage Activity with Duration- Maintenance)
Algorithms (Inherited from parent business object)	• Information: D1-DDACINFO (Device Activity with Duration Information) • Pre-Processing: D1-DETACTTYP (Determine Activity Type)
Lifecycle (Inherited from parent business object)	• Pending (Initial) • Started (Interim) • Ended (Final)

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

Smart Grid Gateway Base Package Command Activity Business Objects

The Oracle Utilities Smart Grid Gateway base package includes the following command activity business objects.

Business Object Name	Description
D1-BulkRequestHeader	Bulk Request Header Instances of this business object represent individual bulk requests headers in the system. Used with bulk commands.
D1-BulkResponse	Bulk Response Instances of this business object represent individual bulk responses in the system. Used with bulk commands.

Business Object Name	Description
D1-CancelCommand	Cancel Command Instances of this business object represent individual command cancellations in the system.
D1-DeviceCommission	Device Commission Instances of this business object represent individual device commission commands initiated in the system.
D1-DeviceDecommission	Device Decommission Instances of this business object represent individual device decommission commands initiated in the system.
D1-DeviceStatusCheck	Device Status Check Instances of this business object represent individual device status check (ping) commands initiated in the system.
D1-GenericConnect	Generic Connect Device
D1-GenericDisconnect	Generic Disconnect Device
D1-GenericReadDevice	Generic Read Device
D1-MultiDeviceStatusCheck	Multi-Device Status Check Instances of this business object represent individual multi-device status check (ping) commands initiated in the system. This differs from the standard Device Status Check in that it will be processed against a list of devices and potentially a list of recipients
D1-OnDemandReadAbstract	On-Demand Read Abstract Parent (used a a parent for on-demand read activity business objects)
D1-OnDemandReadInterval	On-Demand Read Interval Instances of this business object represent individual interval on-demand read commands initiated in the system.
D1-OnDemandReadScalar	On-Demand Read Scalar Instances of this business object represent individual scalar on-demand read commands initiated in the system.
D1-RemoteConnect	Remote Connect Instances of this business object represent individual remote connect commands initiated in the system.
D1-RemoteDisconnect	Remote Disconnect Instances of this business object represent individual remote disconnect commands initiated in the system.

Smart Grid Gateway Base Package Payload Statistics Activity Business Objects

The Oracle Utilities Smart Grid Gateway base package includes the following upload statistics activity business objects.

Business Object Name	Description
D1-PayloadErrorNotif	Payload Error Notification Instances of this business object represent individual payload error notifications in the system.
D1-PayloadExtractScheduler	Payload Extract Scheduler (used a a parent for vendor-specific event and usage extract activity business objects)
D1-PayloadNotification	Payload Notification Instances of this business object represent individual payload notifications in the system.
D1-PayloadStatistics	Payload Statistics Instances of this business object represent individual payload statistics in the system.
D1-PayloadSummary	Payload Summary Instances of this business object represent individual payload statistics summaries in the system.

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.

Configuration Options

This section outlines specific types of BO Options used by Oracle Utilities Smart Grid Gateway command activity business objects.

Business Object Options

Command activity business objects used by Oracle Utilities Smart Grid Gateway can make use of the following business object options:

- **Initiate Command Processing Role**: This option defines the processing role used to initiate the command for the activity. Valid values are defined as values for the PROC_ROLE_FLG lookup field. The base package includes the following processing roles:

Lookup FIeld Value	Description
D1AM	Obtain AMI Device Identifier
D1AR	Appointment Request
D1BR	Bulk Response
D1CA	Cancelation Activity

Lookup FIeld Value	Description
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)
D1SD	Send Device Event
D1SN	Suppression Notification
D1SQ	SQI Translation
D1SR	Response - Success
D1TU	TOU Translation
D1TZ	Time Zone Translation

Lookup Fleld Value	Description
D1UM	UOM Translation
D1UP	Update Activity
D1VC	Verify Commission

Inbound Communications In Detail

This section provides details concerning the inbound communication 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 inbound communication objects you create as part of your implementation. This section includes:

- A description of the D1-COMMIN maintenance object
- Lists of the base package inbound communication business objects, including "lite" business objects
- A sample inbound communication business object (D1-CommInLite)

Maintenance Object - D1-COMMIN

Inbound communication business objects use the D1-COMMIN maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-COMMIN
Description	Communication In
Service Name	D1-COMMIN (Communication In Maintenance)
Tables	• D1_COMM_IN (Communication In) - Primary • D1_COMM_IN_CHAR (Communication In Characteristics) - Child • D1_COMM_IN_IDENTIFIER (Communication In Identifier) - Child • D1_COMM_IN_LOG (Communication In Log) - Child • D1_COMM_IN_LOG_PARM (Communication In Log Parameter) - Child • D1_COMM_IN_REL_OBJ (Communication In 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 Inbound Communication Business Objects

The Service and Measurement Data Foundation base package includes the following "lite" inbound communication business objects:

Business Object Name	Description
D1-CommInLite	Inbound Communication Lite

Oracle Utilities Smart Grid Gateway adapters include vendor-specific inbound communication business objects.

Example Inbound Communication - D1-CommInLite

The table below lists the details of the D1-CommInLite inbound communication business object.

Option	Description
Business Object	D1-CommInLite
Description	Inbound Communication Lite
Maintenance Object	D1-COMMIN (Communication In)
Application Service	F1-DFTAPS (Default Execution Application Service)
Instance Control	Allow New Instances

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

Outbound Communications In Detail

This section provides details concerning the outbound communication 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 outbound communication objects you create as part of your implementation. This section includes:

- A description of the D1-COMMOUT maintenance object
- Lists of the base package outbound communication business objects, including “lite” business objects
- A sample inbound communication business object (D1-CommOutLite)

Maintenance Object - D1-COMMOUT

Inbound communication business objects use the D1-COMMOUT maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-COMMOUT
Description	Communication Out
Service Name	D1-COMMOUT (Communication Out Maintenance)
Tables	• D1_COMM_OUT (Communication Out) - Primary • D1_COMM_OUT_CHAR (Communication Out Characteristics) - Child • D1_COMM_OUT_IDENTIFIER (Communication Out Identifier) - Child • D1_COMM_OUT_LOG (Communication Out Log) - Child • D1_COMM_OUT_LOG_PARM (Communication Out Log Parameter) - Child • D1_COMM_OUT_REL_OBJ (Communication Out 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 Outbound Communication Business Objects

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

Business Object Name	Description
D1-CommOutLite	Outbound Communication Lite
D1-CommOutPlacerBO	Communication Out Placer BO
D1-CommunicationLite	Communication Lite

Oracle Utilities Smart Grid Gateway adapters include vendor-specific outbound communication business objects.

Example Outbound Communication - D1-CommOutLite

The table below lists the details of the D1-CommOutLite outbound communication business object.

Option	Description
Business Object	D1-CommOutLite
Description	Outbound Communication Lite
Maintenance Object	D1-COMMOUT (Communication Out)
Application Service	F1-DFTAPS (Default Execution Application Service)
Instance Control	Allow New Instances

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

Bulk Commands in Detail

This section provides details concerning the bulk command 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 bulk command objects you create as part of your implementation. This section includes:

- **Service and Measurement Data Foundation Base Package Bulk Command Business Objects**
- **Example Activity - D1-BulkRequestHeader**
- **Bulk Command Options**

Service and Measurement Data Foundation Base Package Bulk Command Business Objects

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

Business Object Name	Description
D1-BulkRequestHeader	The bulk request header business object tracks and coordinates the creation of responses to a set of commands requested for a list of devices. While this BO does not retain the list of devices (since this list can be long), its ID is linked to each individual command. The Bulk Request Header's display map shows the current state of progress of its individual commands.
D1-BulkResponse	The bulk response business object constructs an XML list containing responses to individual commands from a bulk request. An empty bulk response is created by individual commands linked to a bulk request as they are completed or discarded. The bulk request header has the responsibility of transitioning the bulk response to assemble its contents and send its message.
D1-OutboundMessageBulk	The outbound message business object used for sending a response for the bulk command.

Example Activity - D1-BulkRequestHeader

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

Option	Description
Business Object	D1-BulkRequestHeader
Description	Bulk Request Header
Maintenance Object	D1-ACTIVITY (Activity)
Application Service	D1-BULKREQBOAS (Bulk Request Header BO)
Instance Control	Allow New Instances
Options	• Related Administration BO: D1-BulkRequestHeaderType (Bulk Request Header Type) • Display UI Map: D1-BulkRequestHeaderDisp (Bulk Request Header - Display) • Portal Navigation Option: d1ActivityNavOpt (Activity Navigation Option) • Display Map Service Script: D1-BuReDtls (Bulk Request Header - Details for Display) • Maintenance UI Map: D1-BulkRequestHeaderMaint (Bulk Request Header - Maintenance)
Algorithms	• Information: D1-CRAINFO (Command Request Activity Information) • Pre-Processing: D1-DETACTTYP (Determine Activity Type) • Pre-Processing: D1-DMRO (Default Measurement Requested Override) • Pre-Processing: D1-VALBULKID (Validate External Bulk Request ID) • Validation: D1-BINPVAL (Bulk Request Input Validation
Lifecycle	• Pending (Initial) • Validate (Interim) • Validation Error (Interim) • Discarded (Interim) • Waiting for Effective Date (Interim) • Awaiting Completion (Interim) • Send Response (Interim) • Completed (Final)

Bulk Command Options

This section outlines specific configuration options that are used by bulk command business objects, including:

- **Algorithms**
- **Lookups**
- **XAI Inbound Services**

Algorithms

The base package includes the following algorithms associated with bulk commands

Algorithm	Description	Where Used/Referenced
D1-BINPVAL	Bulk Request Input Validation	Validation algorithm on D1-BulkRequestHeader
D1-BULKCANC	Cancel Outstanding Individual Commands	Enter algorithm on Discarded state and Completed state of D1-BulkRequestHeader
D1-BULKWAIT	Bulk Awaiting Completion	Monitor algorithm on Awaiting Completion state of D1-BulkRequestHeader
D1-CANBULRES	Cancel Outstanding Bulk Responses	Enter algorithm on Discarded state and Completed state of D1-BulkRequestHeader
D1-PREPBULK	Prepare Bulk Response	Enter algorithm on Completed state and Discarded state of device commands.
D1-SENBULCON	Send Bulk Remote Connect Response	Enter algorithm on Sending Message state of D1-BulkResponse
D1-SENBULDCN	Send Bulk Remote Disconnect Response	Enter algorithm on Sending Message state of D1-BulkResponse
D1-SENBULDSC	Send Bulk Device Status Check Response	Enter algorithm on Sending Message state of D1-BulkResponse
D1-SENBULFIM	Send Bulk Final Measurement Response	Enter algorithm on Sending Message state of D1-BulkResponse
D1-SENBULMES	Send Bulk Message	Enter algorithm on Sending Message state of D1-BulkResponse.
D1-SENBULODR	Send Bulk On-Demand Read Response	Enter algorithm on Sending Message state of D1-BulkResponse
D1-SENBULRES	Send Outstanding Bulk Responses	Enter algorithm on Send Response state of D1-BulkRequestHeader

Algorithm	Description	Where Used/Referenced
D1-SENBULSTM	Send Bulk Start Measurement Response	Enter algorithm on Sending Message state of D1-BulkResponse
D1-SENBULSUC	Send Bulk Command Success Response	Enter algorithm on Sending Message state of D1-BulkResponse
D1-VALBULKID	Validate External Bulk Request ID	Preprocessing algorithm on D1-BulkRequestHeader

Lookups

A monitor algorithm evaluates the response creation option selected on the bulk request activity. The base package includes the following options for the response creation method

Lookup FIeld Value	Description
D1ON	One-by-One
D1TM	Timed
D1WT	Wait for All

XAI Inbound Services

The base package includes the following XAI inbound services associate with bulk commands.

Description	XAI Inbound Service Name	Schema Name
Bulk Request Header	D1-BulkRequestHeader	D1-BulkRequestHeader
Bulk Request Update	D1-BulkRequestUpdate	D1-BULKUPD
Bulk Response	D1-BulkResponse	D1-BulkResponse

Completion Events In Detail

This section provides details concerning the completion 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 completion event objects you create as part of your implementation. This section includes:

- A description of the D1-CEVT maintenance object
- Lists of the base package completion event business objects
- A sample completion event business object (D1-ConnectDevice)

Maintenance Object - D1-CEVT

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

Option/Field	Description
Maintenance Object	D1-CEVT
Description	Completion Event
Service Name	D1-CEVT (Completion Event Maintenance)
Tables	• D1_COMPL_EVT (Completion Event) - Primary • D1_COMPL_EVT_CHAR (Completion Event Characteristics) - Child • D1_COMPL_EVT_LOG (Completion Event Log) - Child • D1_COMPL_EVT_LOG_PARM (Completion Event Log Parameters) - Child • D1_COMPL_EVT_REL_OBJ (Completion Event Related Objects) - 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 Completion Event Business Objects

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

Business Object Name	Description
D1-CommissionDevice	Commission Device Completion Event Instances of this business object represent individual commission device completion events in the system.
D1-CompletionEvent	Completion Event (generic completion event business object used as a parent business object for all other completion event business objects)

Business Object Name	Description
D1-ConnectDevice	Connect Device Completion Event Instances of this business object represent individual connect device completion events in the system.
D1-CreateIMD	Create IMD Completion Event Instances of this business object represent individual create IMD completion events in the system.
D1-DecommissionDevice	Decommission Device Completion Event Instances of this business object represent individual decommission device completion events in the system.
D1-DisconnectDevice	Disconnect Device Completion Event Instances of this business object represent individual disconnect device completion events in the system.

Example Completion Event - D1-ConnectDevice

The table below lists the details of the D1-ConnectDevice completion event business object.

Option	Description
Business Object	D1-ConnectDevice
Description	Connect Device
Maintenance Object	D1-CEVT (Completion Event)
Parent Business Object	D1-CompletionEvent (Completion Event)
Lifecycle Business Object	Completion Event
Application Service	D1-CEVTBOAS (Completion Event BO)
Instance Control	Allow New Instances
Options	• Display UI Map: D1-ConnectDeviceCEvtDisp (Connect Device Completion Event - Display) • Portal Navigation Option: d1cevtTabMenu (Completion Event) • Display Map Service Script: D1-D1-CEvtRetDt (Completion Event - Retrieve Details for Display) • Maintenance UI Map: D1-ConnectDeviceCEvtMaint (Connect Dvc Completion Event - Maintenance)
Algorithms	• Validation: D1-VALTRCEVT (Validate Transition Completion Events)

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

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	• 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	• 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)	• 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)	• Pending (Initial) • Additional Processing (Interim) • Help (Interim) • Discarded (Final) • Processed (Final)

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

Configuring Activities and Device Events

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 Device Communication and Device Event Objects

This section provides high-level overviews of the steps involved in configuring custom command activities, communications (inbound and outbound), and completion 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 Command 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:

   System Events: Create algorithms for the following system events:

   - Applicable Processing Role(s)
3. Set up admin records that define the activity types you will use in your implementation.

Configuring Custom Inbound Communications

Configuring custom inbound communications involves the following steps:

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

Configuring Custom Outbound Communications

Configuring custom outbound communications involves the following steps:

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

Configuring Custom Completion Events

Configuring custom completion events involves the following steps:

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

   Note: Completion event business objects should reference D1-CompletionEvent as their Parent Business Object.

Configuring Bulk Command Processing

This section provides high-level overviews of the steps involved in configuring bulk command activities, communications (inbound and outbound), completion events, and device events.

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 Service Providers

Configuring a service provider associated with bulk commands involves the following steps:

1. Create a service provider using the External Application business object. The external system associated with the service provider should be the bulk command external system. See **Configuring External Systems** on page 7-42 for more information.
2. Add the following processing methods to the service provider:
   - Bulk Response
   - Response - Fail
   - Response - Received
   - Response - Success

Configuring XAI Senders

Configuring the external system involves creating the following outbound message types:

- Bulk Complete Notification
- Bulk Received Notification
- Bulk Response
- Bulk Response Notification
- Bulk Response

Configuring External Systems

The external system should have the same name as the service provider. Configuring the external system involves adding the following outbound message types:

- Bulk Completion Notification
- Bulk Received Notification
- Bulk Response Notification

Sending a Bulk Request from an External Application

You can use a BPEL process to send a bulk request from an external application. Use the following values to send the bulk request:

Parameter	Value
WSDL	http://**<SERVER>**:**<SERVER_PORT>**/soa-infra/services/<adapter>*/BulkRequest/BulkRequestService?WSDL
Endpoint URL	http://**<SERVER>**:**<SERVER_PORT>**/soa-infra/services/<adapter>*/BulkRequest/BulkRequestService

*indicates the specific adapter for which bulk commands are being sent.

Input Arguments

The following XML can be used as a sample for the input arguments to the BPEL process. In this example, the response creation method is D1WT (Wait for All). The external system ID should be the same as the service provider:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/">
  <soap:Body xmlns:ns1="http://xmlns.oracle.com/ouaf/ssn">
    <ns1:bulkRequest>
      <ns1:responseCreationMethod>D1WT</ns1:responseCreationMethod>
      <ns1:timedResponseBeforeAllReceived>false</
ns1:timedResponseBeforeAllReceived>
      <ns1:externalBulkRequestId>ZZ_VK_1003</
ns1:externalBulkRequestId>
      <ns1:externalRequesterId>ZZBULK_XAI</ns1:externalRequesterId>
      <ns1:commandIdentifier>OnDemandReadScalar</
ns1:commandIdentifier>
      <ns1:utilityDeviceIds>
        <ns1:utilityDeviceIdentifierNumber>AM_ODR_SCALAR_01</
ns1:utilityDeviceIdentifierNumber>
        <ns1:utilityDeviceIdentifierNumber>AM_ODR_SCALAR_02</
ns1:utilityDeviceIdentifierNumber>
        <ns1:utilityDeviceIdentifierNumber>AM_ODR_SCALAR_03</
ns1:utilityDeviceIdentifierNumber>
        <ns1:utilityDeviceIdentifierNumber>AM_ODR_SCALAR_04</
ns1:utilityDeviceIdentifierNumber>
      </ns1:utilityDeviceIds>
      <ns1:expirationDateTime></ns1:expirationDateTime>
      <ns1:callbackURL></ns1:callbackURL>
    </ns1:bulkRequest>
  </soap:Body>
</soap:Envelope>
```

The following table describes the elements in the input arguments:

Element	Description
bulkRequest	Header element that contains a bulk request.

Element	Description
responseCreationMethod	Specifies the method in which responses are created for requests submitted from external systems. Valid values include: • One by One (D1ON) • Timed (D1TM) • Wait for All (D1WT)
timedResponseBeforeAllReceived	Specifies whether or not a response should be sent back to external system before all commands are received. Valid values are: • true • false
externalBulkRequestId	An ID given by the external system to represent the request.
externalRequesterId	The ID of the external system submitting the request. This must match the External System linked to the Service provider that represents the external system
commandIdentifier	Specifies the command to initiate for the listed devices. Valid values include: • DeviceStatusCheck • DeviceCommissioning • DeviceDecommissioning • EnableService • OnDemandReadInterval • OnDemandReadScalar • RemoteConnect • RemoteDisconnect
utilityDeviceIds	Element containing one or more utilityDeviceIdentifierNumber elements that specify individual devices for which the specified command is to be initiated.
utilityDeviceIdentifierNumber	The Device Identifier Number for the device for which the specified command is to be initiated. This value should be based on the “AMI Device ID Type” specified on the Service Provider that represents the external system submitting the request.
expirationDateTime	The date and time after which commands that have not yet completed are discarded
callbackURL	The URL from the external system where information is sent if a fault occurs while processing commands. If empty, no fault information is sent.

Chapter 8

Batch Processing

This chapter describes the base package batch processes provided with Oracle Utilities Service and Measurement Data Foundation.

Refer to the following documentation for more information about batch processing:

- Oracle Utilities Smart Grid Gateway 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 Base Package 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	Install Event MO Periodic Monitor Process 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. 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.
D1-MC	MC MO Periodic Monitor Process 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. 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.
D1-MSRMT	Measurement - Derive Other Values 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. 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.
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	Related MC Consumption Sync 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.
D1-RMCRR	Related MC Consumption Sync - Retry Activities 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.
D1-SMMTR	Smart Meter State Monitor Process

Batch Control	Name / Description
D1-SP	Service Point MO Periodic Monitor Process 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. 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.
D1-SYNC	Data Sync MO Periodic Monitor Process
D1-TOUTR	Time of Use Map Data Generation Monitor Invokes monitor algorithms for Time Of Use Map instances whose current state does not reference a monitor process.
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-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.
D1-SRSDE	Snyc 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 a 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.

Chapter 9

Integrating Oracle Utilities Smart Grid Gateway with Other Systems

This chapter provides an overview of how Oracle Utilities Smart Gird Gateway can be integrated with other systems, including:

- **Understanding Communications Processing**
- **Invoking Meter Commands From an External System**
- **Creating Custom Smart Grid Gateway Adapters**
- **Initial Measurement and Device Event XML Formats**
- **Usage and Event Import - Message Driven Bean Configuration**
- **Integrating with an Outage Management System**

Understanding Communications Processing

This section describes the manner in which Oracle Utilities Smart Grid Gateway communicates with head-end systems, including:

- **One-Way Communications: Import of Usage and Events**
- **Two-Way Communications: Meter Commands**

One-Way Communications: Import of Usage and Events

Importing of usage readings and device events is supported through one-way communications with a head-end system: the head-end system sends a payload of usage readings or events, and Oracle Utilities Smart Grid Gateway picks up the payload and imports the readings and/or events into the system.

The table below provides an overview of the process that takes place when usage or 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 Service and Measurement Data Foundation objects used in processing that step or operation.

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 head-end system sends a payload of usage readings or device events to middleware components. The device event paylaod contains both standard and paried device events. Oracle Utilities Smart Grid Gateway adapters use Oracle Service Bus. (OSB) as the middleware components for import of usage readings and device events,	
2.	The middleware parses the payload into individual usage readings or device events and then transforms each from the format used by the head-end system into the SGG format.	
3.	The middleware then invokes an XAI Inbound Service for each reading or device event. For usage readings, the middleware invokes the Initial Load IMD inbound service. This service is mapped to the IMD Seeder business object. For device events, the middleware invokes he Device Event Seeder inbound service. This service is mapped to the Device Event Seeder business object.	Usage Reading XAI Inbound Service: D1-InitialLoadIMD (D107437309) Device Event XAI Inbound Service: D1-DeviceEventSeeder (D103879193)

Step	Process	Service and Measurement Data Foundation Objects
4.	The IMD Seeder business object determines the type of initial measurement business object to create for each reading based on the processing method defined for the "Initial Measurement Creation" processing role on the head-end system service provider. Different head-end systems may use different names for units of measure (UOMs). The IMD Seeder business object maps vendor-specific UOMs to "standard" UOMs based on the processing method defined for the "UOM Mapping" processing role on the head-end system service provider. 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. 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.	IMD Seeder Processing Method: D1-HowToCreateMCInformation (How to Create MD Related Information) UOM MappingBO: D1-HowToProcessDeviceInfo (How to Process Device Related Information) Device Event Seeder Processing Method: D1-HowToProcessDeviceInfo (How to Process Device Related Information) Device Event Mapping BO: D1-DeviceEventMappingLookup
5.	An instance of the appropriate initial measurement data business object is created in the system for each usage reading received. An instance of the appropriate device event business object is created in the system for each device event received. For paired events, the first event business objects may create activities to represent the event.	Initial Measurement Data BOs: • D1-InitialLoadIMDInterval • D1-InitialLoadIMDScalar Standard Device Event BO: D1-StandardDeviceEvent Paired (First) Device Event BO: D1-PairedEventFirstDeviceEvent (Device Event - Paired Event (First) Paired (Last) Device Event BO: D1-PairedEventLastDeviceEvent (Device Event - Paired Event (Last)
6.	Device Events Only: 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. Sending device event information is triggered by an Enter algorithm on the "Processed" Status of the device event business object's lifecycle. 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.	Procesing Method: D1-HowToProcDveEvtsInformation (How to Process Device Event Related Info)

Refer to the following documentation for more details concerning the specific configurations of Oracle Service Bus and the specific business objects used by Oracle Utilities Smart Gird Gateway adapters for importing usage reading and device events:

- Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr Configuration Guide
- Oracle Utilities Smart Grid Gateway Adapter for Echelon Configuration Guide
- Oracle Utilities Smart Grid Gateway MV90 Adapter for Itron Configuration Guide
- Oracle Utilities Smart Grid Gateway Adapter for Sensus Configuration Guide
- Oracle Utilities Smart Grid Gateway Adapter for Silver Springs Network Configuration Guide
- Oracle Utilities Smart Grid Gateway Adapter Development Kit Configuration Guide
- Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay Configuration Guide

Two-Way Communications: Meter Commands

Issuing meters commands and receiving a response from the head-end system is supported through two-way communications with a head-end system: Oracle Utilities Smart Grid Gateway sends a message for the command, the head-end system receives and performs the command, and then sends an acknowledgement back to Oracle Utilities Smart Grid Gateway.

The table below provides an overview of the communication process that takes place when a command is initiated for a device. 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 Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr. Refer to the *Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr Configuration Guide* for more details about the configuration objects used by this adapter.

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

Step	Process	Landis+Gyr Adapter Objects
1.	A user initiates a remote connect command for a device. A remote connect activity business object is instantiated for the command.	Activity BO: D1-RemoteConnect (Remote Connect)
2.	The remote connect command activity business object creates an outbound communication. The specific type of outbound communication business object created is determined by the head-end system service provider (based on the processing role defined in the Enter algorithm of the “Awaiting Response” status of the outbound communication business object’s lifecycle).	Outbound Comunication BO: D3-InitiateConnectDisconnect (Initiate Connect Disconnect)
3.	The outbound communication creates an outbound message. The specific type of outbound message created is determined by the head-end system service provider. (based on the processing role defined in the Enter algorithm of the “Awaiting Response” status of the outbound communication business object’s lifecycle).	Outbound Message Type: D3-CONNECT (Connect Device)

Step	Process	Landis+Gyr Adapter Objects
4.	The outbound message is sent to middleware components via an External System and XAI Sender. Oracle Utilities Smart Grid Gateway adapters use Oracle Business Process Execution Language (BPEL) as the middleware components for communication with head-end systems.	External System: Landis+Gyr Head End System XAI Sender: D3-Connect (Connect Device)
5.	The middleware converts the outbound message from SGG format into the format used by the head-end system, and sends the message to the head-end system.	
6.	When the head-end system sends a response, the middlware receives the response message from the head-end system, and converts it from the format used by the head-end system to SGG format and invokes an XAI Inbound Service.	XAI Inbound Service: D3-ConDisconStChgNotification (D361040925)
7.	The XAI Inbound Service picks up the message, and creates a corresponding inbound communication. The specific type of inbound communication business object created is determined by the XAI Inbound Service.	XAI Inbound Service: D3-ConDisconStChgNotification (D361040925) Inbound Communication BO: D3-ConnectDisconStateChgNtf (Connect Disconnect State Changed Notification (Multispeak))
8.	The inbound communication identifies the parent outbound communication.	Outbound Comunication BO: D3-InitiateConnectDisconnect (Initiate Connect Disconnect)
9.	The inbound communication creates a completion event to update the status of the device to indicate it has been connected. The specific type of completion event business object created is specified in an Enter algorithm on the “Create Completion Event” Status of the inbound communication business object’s lifecycle.	Completion Event BO: D1-ConnectDevice (Connect Device) Algorithm: D3-CCE (Create Completion Event)
10.	The inbound communication updates the outbound communication. This update is performed by an Enter algorithm on the “Completed” Status of the inbound communication business object’s lifecycle.	Outbound Comunication BO: D3-InitiateConnectDisconnect (Initiate Connect Disconnect)
11.	The outbound communication updates the “Connect/Disconnect Completion Flag” and the original activity business object. This update is performed by an Enter algorithm on the “Completed” Status of the outbound communication business object’s lifecycle.	Outbound Comunication BO: D3-InitiateConnectDisconnect (Initiate Connect Disconnect)

Refer to the following documentation for more details concerning the specific configurations of Oracle Business Process Execution and the specific business objects used by Oracle Utilities Smart Gird Gateway adapters for supporting meter commands via two-way communication:

- Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr Configuration Guide
- Oracle Utilities Smart Grid Gateway Adapter for Echelon Configuration Guide
- Oracle Utilities Smart Grid Gateway Adapter for Sensus Configuration Guide
- Oracle Utilities Smart Grid Gateway Adapter for Silver Springs Network Configuration Guide
- Oracle Utilities Smart Grid Gateway Adapter Development Kit Configuration Guide
- Oracle Utilities Smart Grid Gateway Adapter for Itron OpenWay Configuration Guide

Invoking Meter Commands From an External System

This section provides an overview of how to issue or invoke commands through Oracle Utilities Smart Grid Gateway from an external system. This is useful if implementing Oracle Utilities Smart Grid Gateway alongside external systems for which no productized integration is available, such as an outage system or customer information system.

This section includes:

- **Defining External System as a Service Provider**
- **Exposing Command Activity Business Objects**
- **Sending Command Responses to the External System**
- **Configuring A Middleware Communication Layer**

Defining External System as a Service Provider

The first step in enabling the issuing of commands through Oracle Utilities Smart Grid Gateway from an external system is to define the external system as a service provider. Several processes within Oracle Utilities Smart Grid Gateway use the service provider and its associated processing methods to determine how to process specific types of operations.

Service Providers are defined by the following:

- **Service Provider Business Object**: The business object used to create the service provider. For external systems, this should be "External Application (D1-ExternalApplication)."
- **Service Provider**: A code that designates the external system
- **Description**: A description of the external system
- **External Reference ID**: A reference ID for the external system
- **External System**: A related external system (if applicable)
- **Our Name/ID in Their System**: The name of Oracle Utilities Smart Grid Gateway as defined in the external system
- **Utility Device ID Type**: The type of ID used by the external system to identify devices (badge number, external ID, internal meter number, serial number, or pallet number)
- **Utility Measuring Component ID Type**: The type of ID used by the external system to identify measuring components (channel ID or external ID)

Refer to the Oracle Utilities Service and Measurement Data Foundation user documentation for more information about creating service providers.

The table below contains an example service provider.

Attribute/Field	Value
Service Provider Business Object	External Application (D1-ExternalApplication)
Service Provider	CM_OUTAGE_SYSTEM
Description	Outage System
External Reference ID	12345
External System	N/A
Our Name/ID in Their System	OUSGG
Utility Device ID Type	Serial Number

Attribute/Field	Value
Utility Measuring Component ID Type	Channel ID

Processing Methods

You must also define a number of processing methods for the service provider. These designate the method by which command (activity) related information is sent to the external system.

In this case, you want to define processing methods that designate how outbound messages related to commands (activities) are sent. The table below outlines the processing roles and methods that can be used for this.

Processing Role	Processing Method
Response - Received (D1RR)	How to Send Activity Related Outbound Messages (D1-HowToSendActivityResponses)
Response - Success (D1SR)	How to Send Activity Related Outbound Messages (D1-HowToSendActivityResponses)
Response - Failure (D1FR)	How to Send Activity Related Outbound Messages (D1-HowToSendActivityResponses)

For each of the above processing methods, you would define the outbound message type, or message category and message number to use when sending messages. The type of message to send can be a default type, or can be specified by activity (command) type.

Exposing Command Activity Business Objects

The next step in enabling the issuing of commands through Oracle Utilities Smart Grid Gateway from an external system is to expose the activity business objects for the commands you wish to invoke as web services. This is done by creating an XAI Inbound Service for each command.

XAI Inbound Services are defined by the following:

- **XAI Inbound Service Name**: A code for the inbound service.
- **Adapter**: The type of adapter used by the inbound service. This should be “BusinessAdapter (Business Adapter)”.
- **Schema Type**: The type of object to be invoked via the inbound service (business object, business service, or service script). This should be Business Object.
- **Schema Name**: The activity business object to invoke for the command
- **Description**: A description of the inbound service.

When you create an XAI Inbound Service, the system creates the Web Service Definition Language (WSDL) schema for the service. This can be used by the external system to invoke the web services associated with each command.

The table below lists a set of sample XAI Inbound Services that could be created to support the commands supported through Oracle Utilities Smart Grid Gateway.

XAI Inbound Service	Schema Name
CM-DeviceCommissionService	D1-DeviceCommission (Device Commissioning)
CM-DeviceDecommissionService	D1-DeviceDecommission (Device Decommissioning)
CM-DeviceStatusCheckService	D1-DeviceStatusCheck (Device Status Check)

XAI Inbound Service	Schema Name
CM-OnDemandReadIntervalService	D1-OnDemandReadInterval (On-Demand Read Interval)
CM-OnDemandReadScalarService	D1-OnDemandReadScalar (On-Demand Read Scalar)
CM-RemoteConnectService	D1-RemoteConnect (Remote Connect)
CM-RemoteDisconnectService	D1-RemoteDisconnect (Remote Disconnect)

Refer to the Oracle Utilities Application Framework documentation for more information about XAI Inbound Services.

Sending Command Responses to the External System

The next step in enabling the issuing of commands through Oracle Utilities Smart Grid Gateway from an external system is to set up the configuration necessary to send outbound messages from Oracle Utilities Smart Grid Gateway to the external system. This includes configuring the following:

- **Outbound Message Types**
- **XAI Senders**
- **External Systems**

Outbound Message Types

A outbound message type must also be created for each type of message to be sent to an external system, based on the types of messages the system is designed to accept. To continue the example above, you might create the following outbound message types:

Command	Outbound Message Type
Commission Device	Commission Device
Decommission Device	Decommission Device
Device Status Check	Device Status Check
On-Demand Read Interval	On-Demand Read Interval
On-Demand Read Scalar	On-Demand Read Scalar
Remote Connect	Connect Device
Remote Disconnect	Disconnect Device

Refer to the Oracle Utilities Application Framework documentation for more information about outbound message types.

XAI Senders

You must also create an XAI Sender for each type of message to be sent to an external system. XAI senders define the details of how messages are sent to an external system. The set of XAI senders you need to create is based on the types of messages the system is designed to accept. To continue the example above, you might create the following XAI senders:

Command	XAI Sender
Commission Device	Commission Device
Decommission Device	Decommission Device
Device Status Check	Device Status Check
On-Demand Read	On-Demand Read
Remote Connect	Connect Device
Remote Disconnect	Disconnect Device

Refer to the Oracle Utilities Application Framework documentation for more information about XAI senders.

External Systems

You must also create an External System for each external system to which Oracle Utilities Smart Grid Gateway will send messages. Each external system defines a set of outbound message types that will be sent to that system. Each external system outbound message type also specifies the following:

- The processing method used to send the message (Batch, XAI, or Real-time)
- The corresponding XAI senders
- Batch Control (if Processing Method is set to Batch)
- Message XSL, W3C Schema, and Response XSL (as applicable)

To continue the example above, you might create the following external system:

Head-End System			
Outbound Message Type	**Processing Method**	**XAI Sender**	**Message XSL / Response XSL**
Commission Device	Real-time	Commission Device	OS-Request.xsl / OS-Response.xsl
Decommission Device	Real-time	Decommission Device	OS-Request.xsl / OS-Response.xsl
Device Status Check	Real-time	Device Status Check	OS-Request.xsl / OS-Response.xsl
On-Demand Read	Real-time	On-Demand Read	OS-Request.xsl / OS-Response.xsl
Remote Connect	Real-time	Connect Device	OS-Request.xsl / OS-Response.xsl
Remote Disconnect	Real-time	Disconnect Device	OS-Request.xsl / OS-Response.xsl

Refer to the Oracle Utilities Application Framework documentation for more information about external systems.

Configuring A Middleware Communication Layer

The last step in enabling the issuing of commands through Oracle Utilities Smart Grid Gateway from an external system is to configure a middleware component that sends web service requests from the external system to the XAI Inbound Services, and receives messages from the XAI Senders. This can via queue-based messaging such as JMS queues, or via tools such as Oracle SOA Suite and Business Process Execution Language (BPEL).

Creating Custom Smart Grid Gateway Adapters

This section provides an overview of the steps involved in creating custom adapters for use with Oracle Utilities Smart Grid Gateway. Adapters are configurations designed to support the specific communication requirements of a specific head-end system currently not supported by Oracle Utilities Smart Grid Gateway, and include:

- **Adapters for Importing Usage Readings and Device Events**
- **Adapters for Issuing and Initiating Commands**
- **Creating Custom Commands**

The Oracle Utilities Smart Grid Gateway demonstration data includes a "generic" adapter that can be used as the basis for creating custom adapters. See **Appendix D**: **The Oracle Utilities Smart Grid Gateway "Generic" Adapter** for more information about the Oracle Utilities Smart Grid Gateway generic adapter.

Adapters for Importing Usage Readings and Device Events

This section outlines the steps involved in creating an adapter to support import of usage readings and device events from a head-end system. These steps include:

- **Configure Middleware**
- **Create Business Objects**
- **Create Service Provider for Head-End System**

Configure Middleware

Middleware is used to facilitate communication between the head-end system and Oracle Utilities Smart Grid Gateway. In the case of importing usage readings and device events, the middleware performs the following steps:

- Accepts payloads from head-end system

  The middleware layer must be configured to accept payloads from the head-end system. This is often an queue-based process utilizing JMS queues configured within an application server.

- Parses payloads into individual readings/events

  Usage readings and device events are typically send data in payloads comprising hundreds or even thousands of readings or events. The middleware layer parses these payloads into individual readings or event, each of which is then processed separately.

- Transforms readings/events from head-end system format into SGG format

  Usage readings and device events are sent in a format used by the head-end system. The middleware layer transforms each reading or event from the head-end system format into the XML format used by Oracle Utilities Smart Grid Gateway. See **Initial Measurement and Device Event XML Formats** on page 9-31 for details concerning this XML format.

- Sends each reading or event to Oracle Utilities Smart Grid Gateway by invoking an XAI Inbound Service

  After parsing and transforming the incoming data, the middleware then sends each reading or event to the application. This can be done via web service call to an XAI Inbound Service or by depositing the reading or event in a JMS queue.

- Creates upload statistics activities to record the processing of each payload

  As it processes each payload (including all the steps above), the middleware creates upload statistics activities as follows:

- When processing begins for a payload, a Payload Statistics activity is created to record the process.
- If an error occurs during processing, a Payload Error Notification activity is created.
- When payload processing is complete, a Payload Summary activity is created, which in turn, updates the Payload Statistics activity with details concerning the processing of the payload.

See **Upload Statistics** on page 7-5 for more information about upload statistics activities.

Oracle Utilities Smart Grid Gateway adapters use Oracle Service Bus (OSB) to perform these operations. Refer to the following documentation for more details concerning the specific configurations of Oracle Service Bus used by Oracle Utilities Smart Gird Gateway adapters for importing usage reading and device events:

- Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr Configuration Guide
- Oracle Utilities Smart Grid Gateway MV90 Adapter for Itron Configuration Guide
- Oracle Utilities Smart Grid Gateway Adapter for Echelon NES Configuration Guide

Pre-configured OSB processes can be extended to support custom requirements during processing.

Create Business Objects

For each type of usage reading and/or device event to be imported, you need to create business object that defines the initial measurement or device event. If the head-end system uses specific device event names, units of measure, or condition codes, you may also need to create extendable lookup business objects that can be used to perform mapping between the head-end system codes and “standard” codes.

Initial Measurement Data Business Objects

Initial measurement data business objects defines the types of initial measurement data to be received from the head-end system. You must create an initial measurement data business object for each type of initial measurement data you expect to store. For example, the Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr uses the following initial measurement data business objects:

Initial Measurement Data Business Object	Description
D3-InitialLoadIMDInterval (Landis+Gyr Initial Load IMD - Interval)	Used for interval initial measurement data received from Landis+Gyr.
D3-InitialLoadIMDScalar (Landis+Gyr Initial Load IMD - Scalar)	Used for scalar initial measurement data received from Landis+Gyr.

Notes:

- All head-end system-specific inteval initial measurement data business objects should be child business objects of the D1-InitialLoadIMDInterval business object.
- All head-end system-specific scalar initial measurement data business objects should be child business objects of the D1-InitialLoadIMDScalar business object.

See **Initial Measurement Data In Detail** on page 6-21 for more information about initial measurement data business objects.

Measurement Business Objects

Measurement business objects represent the different types of final measurements to be stored in the system that originate from the head-end system. If final measurements need to capture head-end system-specific data elements, you should create a measurement business object to accomodate those data elements. See **Measurements In Detail** on page 6-26 for more information about measurement business objects.

Device Event Business Objects

Device event business objects represent the different types of device events that can be sent by the head-end system. 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.

Oracle Utilities Smart Grid Gateway provides a set of basic device event business objects that can extended to support head-end system-specific device events.

- Device events with no duration are defined using "standard" device event business objects.
- 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.

Notes:

- All head-end system-specific device event business objects should be child business objects of the D1-DeviceEvent business object.

See **Device Events In Detail** on page 7-37 for more information about device event business objects.

Extendable Lookup Business Objects

If the head-end system uses its own set of units of measure, or device event names, or condition codes, you can create extendable lookup business objects to perform mappings between the head-end system codes and the standard codes used by Oracle Utilities Smart Grid Gateway. You reference these business objects in processing methods defined for the service provider that represents the head-end system.

Refer to the Oracle Utilities Application Framework documentation for more information extendable lookups.

Create Service Provider for Head-End System

Each head-end system that communicates with Oracle Utilities Smart Grid Gateway should be defined as a service provider. The service provider's processing roles and methods are used by many processes to determine the specific manner in which data is to be sent or received from the service provider. For example, the "Initial Measurement Creation" processing role defines the specific type of initial measurement data business object to use when creating initial measuremenst received from the head-end system.

For example, the Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr could use the following service provider to represent the Landis+Gyr Command Center head-end system.

Attribute/Field	Value
Service Provider Business Object	Head-End System (D1-HeadEndSystem)
Service Provider	LG
Description	Landis+Gyr
External Reference ID	LG
External System	Landis+Gyr
Our Name/ID in Their System	
AMI Device ID Type	Internal Meter Number

Attribute/Field	Value
AMI Measuring Component ID Type	Channel ID

This service provider uses the following processing methods:

Processing Role	Processing Method
Initial Measurement Creation / How to create Initial Load IMD	D1-HowToCreateMCInformation (How To Create MC Related Information) This processing method specifies the type of intial measurement data business object to use when creating initial measurements received from Landis+Gyr. (D3-InitialLoadIMDScalar) Note: The processing methods can define a default initial measurement data business object, or can specify an initial measurement data business object for each Measuring Component Type.
UOM Mapping / How to map UOMs from L+G when creating IMDs	D1-HowToProcessDeviceInfo (How to Process Device Related Information) This processing method specifies the extendable lookup business object to use when mapping Landis+Gyr UOM codes to SGG UOM codes. (D3-HeadendUOMLookup) Note: The processing methods can define a default extendable lookup business object, or can specify an extendable lookup business object for each Device Type.

Note: The table above only lists processing methods used in processing readings and device events.

Another processing role and method used in processing device events is described below

Processing Role	Processing Method
Device Event Mapping / How to Map Device Events	D1-HowToProcessDeviceInfo (How to Process Device Related Information) This processing method specifies the extendable lookup business object to use when mapping Landis+Gyr device event names to “standard” device event names. Note: The processing methods can define a default extendable lookup business object, or can specify an extendable lookup business object for each Device Type.

See **Service Providers In Detail** on page 5-20 and **Processing Methods In Detail** on page 5-23 for more information about service providers and processing methods.

Adapters for Issuing and Initiating Commands

This section outlines the steps involved in creating an adapter to support initiating commands and communications between Oracle Utilities Smart Grid Gateway and a head-end system. These steps include:

- **Configure Outbound Communication**
- **Configure Middleware**
- **Configure Inbound Communication**
- **Create Service Provider for Head-End System**
- **Creating Custom Commands**

Configure Outbound Communication

Outbound Communications represent messages sent from Oracle Utilities Smart Grid Gateway to the head-end system or edge application (such as Oracle Customer Care and Billing). Outbound communications use the following types of objects:

- **Outbound Communication Business Objects**
- **Outbound Message Types**
- **XAI Senders**
- **External Systems**

Outbound Communication Business Objects

Outbound communication business object must be created for each type of message to be sent to the head-end system. This is based on the types of messages the system is designed to accept. For example, suppose a head-end system supports the following types of commands: device commission, device decommission, device status check, on-demand readings, remote connect, and remote disconnect, and that this head-end system accepts a separate type of message for each command. For this example, you would need to create outbound communication business objects for each command, as follows:

Command	Outbound Communication Business Object
Commission Device	Commission Device Outbound Communication
Decommission Device	Decommission Device Outbound Communication
Device Status Check	Device Status Check Outbound Communication
On-Demand Read	On-Demand Read Outbound Communication
Remote Connect	Remote Connect Outbound Communication
Remote Disconnect	Remote Disconnect Outbound Communication

The specifics of each business object are based on the type of message to be sent to the head-end system. The table below provides an example of the “Initiate Connect Disconnect” outbound communication business object used by the Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr.

Option	Description
Business Object	D3-InitiateConnectDisconnect
Description	Initiate Connect Disconnect
Maintenance Object	D1-COMMOUT (Communication Out)
Application Service	D3-INITIATECONDISCON (Initiate Connect Disconnect)
Instance Control	Allow New Instances
Options	• Related Administration BO: D3-InitiateConnectDisconType (Initiate Connect Disconnect Type) • Display UI Map: D3-InitConDisconMltSpkDisplay (Initiate Connect Disconnect Display) • Portal Navigation Option: d1OBCommunicationNavOpt (Outbound Communication Navigation Option) • Display Map Service Script: D3-RtICDDtls (Connect Disconnect - Retrieve Details for Display) • Maintenance Map UI Map: D3-InitConDisconMltSpkMaint (Connect Disconnect Maintenance)
Algorithms	• Information: D1-COMMINFO (Communication Information) • Pre-Processing: D3-GDRFFPA (Default required fields from Parent Activity) • Pre-Processing: D3-DLAC (Default Load Action Check)

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Lifecycle / Algorithms

Note: Does not include “Transition to Default Next Status” monitor algorithms.</td><td>Pending
Validate
Enter: D1-VALCOMTP (Validate Communication Type)
Validation Error
Monitor: D1-RBOE (Rety BO in Error)
Enter: D1-CTDEBOE (Create To Do Entry for BO in Error)
Exit: D1-GTDCBO (Generic To Do Completion for BOs)
Awaiting Response
Monitor: D1-TIMEOUT (Time Out)
Enter: D3-PICDSD (Populate Initial Connect Disconnect Send Detail)
Enter: D3-CCDOBTMSG (Create Connect/Disconnect Out Bound Message)
Retry
Response Error
Same as Validation Error (above)
Discarded
Enter: D1-TPATOF (Transition Parent Activity to Failed)
Completed
Enter: D3-UCCDFTPA (Update Connect Disconnect Completion Flag And Transition Parent Activity)</td></tr>
</table>

Outbound Message Types

A outbound message type must also be created for each type of message to be sent to the head-end system. Again, this is based on the types of messages the head-end system is designed to accept. To continue the example above, you might create the following outbound message types:

Command	Outbound Message Type
Commission Device	Commission Device
Decommission Device	Decommission Device
Device Status Check	Device Status Check
On-Demand Read	On-Demand Read
Remote Connect	Connect Device
Remote Disconnect	Disconnect Device

Refer to the Oracle Utilities Application Framework documentation for more information about outbound message types.

XAI Senders

You must also create an XAI Sender for each type of message to be sent to an external system. XAI senders define the details of how messages are sent to an external system. As in the case of outbound communication business objects and outbound message types, the set of XAI senders you need to create is based on the types of messages the head-end system is designed to accept. To continue the example above, you might create the following XAI senders:

Command	XAI Sender
Commission Device	Commission Device
Decommission Device	Decommission Device
Device Status Check	Device Status Check
On-Demand Read	On-Demand Read
Remote Connect	Connect Device
Remote Disconnect	Disconnect Device

Refer to the Oracle Utilities Application Framework documentation for more information about XAI senders.

External Systems

You must also create an External System for each external system to which Oracle Utilities Smart Grid Gateway will send messages. Each external system defines a set of outbound message types that will be sent to that system. The external system also specifies the following for each outbound message type:

- The outbound message type
- The processing method used to send the message (Batch, XAI, or Real-time)
- The corresponding XAI senders
- Batch Control (if Processing Method is set to Batch)
- Message XSL, W3C Schema, and Response XSL (as applicable)

To continue the example above, you might create the following external system:

Head-End System			
Outbound Message Type	**Processing Method**	**XAI Sender**	**Message XSL / Response XSL**
Commission Device	Real-time	Commission Device	HES-Request.xsl / HES-Response.xsl
Decommission Device	Real-time	Decommission Device	HES-Request.xsl / HES-Response.xsl
Device Status Check	Real-time	Device Status Check	HES-Request.xsl / HES-Response.xsl
On-Demand Read	Real-time	On-Demand Read	HES-Request.xsl / HES-Response.xsl
Remote Connect	Real-time	Connect Device	HES-Request.xsl / HES-Response.xsl

Head-End System			
Outbound Message Type	**Processing Method**	**XAI Sender**	**Message XSL / Response XSL**
Remote Disconnect	Real-time	Disconnect Device	HES-Request.xsl / HES-Response.xsl

Refer to the Oracle Utilities Application Framework documentation for more information about external systems.

Configure Middleware

Middleware is used to facilitate communication between the head-end system and Oracle Utilities Smart Grid Gateway. In the case of processing commands, the middleware performs the following steps:

- Accepts outbound messages from XAI Senders

  The middleware must monitor the XAI senders configured to send messages from Oracle Utilities Smart Grid Gateway to the head-end system.

- Transforms communications from SGG format into head-end system format

  Messages are sent in a standard format used by Oracle Utilities Smart Grid Gateway. The middleware must transform each message from this standard format into the format used by the head-end system.

- Sends messages to the head-end system

  The middleware must then send the transformed message to the head-end system. In Oracle Utilities Smart Grid Gateway adapters, this is done via a web service call to web services defined for the head-end system.

- Receives responses from the head-end system

  The middleware also receives responses from the head-end system that indicate if the command was successfully carried out.

- Transforms communications from head-end system format into SGG format

  Response messages are sent in a format used by the head-end system. The middleware must transform each message from this format into the standard format used by Oracle Utilities Smart Grid Gateway.

- Sends response message to Oracle Utilities Smart Grid Gateway by invoking an XAI Inbound Service

  The middleware then sends the response message to the application. This can be done via web service call to an XAI Inbound Service or by depositing the message in a JMS queue.

Oracle Utilities Smart Grid Gateway adapters use Oracle Business Process Exeecution Language (BPEL) to perform these operations. Refer to the following documentation for more details concerning the specific configurations of BPEL used by Oracle Utilities Smart Gird Gateway adapters for processing commands:

- Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr Configuration Guide

Pre-configured BPEL processes can be extended to support custom requirements during message processing.

Configure Inbound Communication

Inbound Communications represent messages sent from a head-end system or edge application (such as Oracle Customer Care and Billing) to Oracle Utilities Smart Grid Gateway. Inbound communications are typically sent in response to a command. Inbound communications use the following types of objects:

- **Inbound Communication Business Objects**
- **XAI Inbound Services**

Inbound Communication Business Objects

An inbound communication business object must be created for each type of message to be received from the head-end systems, based on the types of messages the system is designed to send. To continue the above example, a head-end system that supports the types of commands outlined above (device commission, device decommission, device status check, on-demand readings, remote connect, and remote disconnect) might send a separate type of message in response to each command. For this example, you would need to create the following inbound communication business objects:

Command Being Responded To	Inbound Communication Business Object
Commission Device	Commission Device Response
Decommission Device	Decommission Device Response
Device Status Check	Device Status Check Response
On-Demand Read	On-Demand Read Response
Remote Connect	Remote Connect Response
Remote Disconnect	Remote Disconnect Response

The specifics of each business object are based on the type of message to be sent to the head-end system. The table below provides an example of the "Connect Disconnect State Changed Notification (Multispeak)" inbound communication business object used by the Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr.

Option	Description
Business Object	D3-ConnectDisconStateChgNtf
Description	Connect Disconnect State Changed Notification (Multispeak)
Maintenance Object	D1-COMMIN (Communication In)
Application Service	D3-CONDISSTCHGNTBOAS (State Change Notification BO)
Instance Control	Allow New Instances

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Options</td><td>Related Administration BO: D3-ConnectDisconStateChgNtfTyp (Connect Disconnect State Change Notify Type)
Display UI Map: D3-ConDisconStChgMltSpkDisplay (State Change Notification - Display)
Portal Navigation Option: d1IBCommunicationNavOpt (Inbound Communication Navigation Option)
Display Map Service Script: D3-RtSCNDtls (State Change - Retrieve Details for Display)
Maintenance Map UI Map: D3-ConDisconStChgMltSpkMaint (State Change Notification - Maintenance)</td></tr>
<tr><td>Algorithms</td><td>Information: D1-COMMINFO (Communication Information)
Pre-Processing: D3-SETDFBOEC (Set Default BO elements - Connect/Disconnect Notification)</td></tr>
<tr><td>Lifecycle / Algorithms

Note: Does not include “Transition to Default Next Status” monitor algorithms.</td><td>Pending
Validate
Enter: D3-FPCBORC (Find Parent Communication BO Remote Connect)
Enter: D1-VALCOMTP (Validate Communication Type)
Enter: D3-VSCC (Validate State Change Code)
Validation Error
Monitor: D1-RBOE (Rety BO in Error)
Enter: D1-CTDEBOE (Create To Do Entry for BO in Error)
Exit: D1-GTDCBO (Generic To Do Completion for BOs)
Discarded
Enter: D3-FAILPCOUT (Fail Parent Outbound Communication)
Create Completion Event
D3-CCE (Create Completion Event)
Completed
Enter: D3-UPCMEVDTR (Update Parent Communication’s Event Date Time Connect)
Enter: D3-TRANPRBO (Transition Parent Outbound Communication BO)</td></tr>
</table>

XAI Inbound Services

You must also create an XAI Inbound Service for each type of message to be received from an external system. XAI inbound systems define the details of how messages are received from an external system, including the inbound communication business object (or business service or service script) to be invoked when the response message is received. As in the case of inbound communication business objects, the set of XAI inbound services you need to create is based on the types of messages the system is designed to send. To continue the example above, you might create the following XAI inbound services:

XAI Inbound Service	Schema (Inbound Communication Business Object)
Commission Device Response	Commission Device Response
Decommission Device Response	Decommission Device Response
Device Status Check Response	Device Status Check Response
On-Demand Read Response	On-Demand Read Response
Connect Device Response	Remote Connect Response
Disconnect Device Response	Remote Disconnect Response

Refer to the Oracle Utilities Application Framework documentation for more information about XAI Inbound Services.

Create Service Provider for Head-End System

Each head-end system that communicates with Oracle Utilities Smart Grid Gateway should be defined as a service provider. The service provider's processing roles and methods are used by many processes to determine the specific manner in which data is to be sent or received from the service provider. For example, the "Remote Connect" processing role defines the specific type of outbound communication business object to use when sending outbound communications to the head-end system as a result of a remote connect command.

The Landis+Gyr service provider provided with the Oracle Utilities Smart Grid Gateway Adapter for Landis+Gyr uses the following processing methods:

Processing Role	Processing Method
On-Demand Read (Scalar) (D1SC)	D1-HowToCreateActivityOBComm (How To Create OB Communication/Send OB Message) This processing method specifies the type of outbound communication business object and outbound message type to use when sending outbound communications to Landis+Gyr as a result of an on-demand read (scalar) command. (D3-InitiateMRByMtrNbr / Initiate Meter Read By Meter Number (CMD3))

Processing Role	Processing Method
Remote Connect (D1RC)	D1-HowToCreateActivityOBComm (How To Create OB Communication/Send OB Message) This processing method specifies the type of outbound communication business object and outbound message type to use when sending outbound communications to Landis+Gyr as a result of a remote connect command. (D3-InitiateConnectDisconnect / Connect Device (D3-CONNECT))
Remote Disconnect (D1RD)	D1-HowToCreateActivityOBComm (How To Create OB Communication/Send OB Message) This processing method specifies the type of outbound communication business object and outbound message type to use when sending outbound communications to Landis+Gyr as a result of a remote connect command. (D3-InitiateConnectDisconnect / Disconnect Device (D3-DISCONNEC))

Note: The table above only lists processing methods used in processing commands.

Command processing methods can define a default outbound communication business object/ outbound message type, or can specify an outbound communication business object/outbound message type for each Device Type.

Other processing roles and methods used in processing commands are described below

Processing Role	Processing Method
Device Commission (D1DC)	D1-HowToCreateActivityOBComm (How To Create OB Communication/Send OB Message) This processing method specifies the type of outbound communication business object and outbound message type to use when sending outbound communications to a head-end system as a result of a commission device command.
Device Decommission (D1DD)	D1-HowToCreateActivityOBComm (How To Create OB Communication/Send OB Message) This processing method specifies the type of outbound communication business object and outbound message type to use when sending outbound communications to a head-end system as a result of a decommission device command.

Processing Role	Processing Method
Device Status Check (D1DS)	D1-HowToCreateActivityOBComm (How To Create OB Communication/Send OB Message) This processing method specifies the type of outbound communication business object and outbound message type to use when sending outbound communications to a head-end system as a result of a device status check command.
On-Demand Read (Interval) (D1IN)	D1-HowToCreateActivityOBComm (How To Create OB Communication/Send OB Message) This processing method specifies the type of outbound communication business object and outbound message type to use when sending outbound communications to a head-end system as a result of an on-demand read (interval) command.

See **Service Providers In Detail** on page 5-20 and **Processing Methods In Detail** on page 5-23 for more information about service providers and processing methods.

Creating Custom Commands

Oracle Utilities Smart Grid Gateway supports a number of commands, including:

- Commission Device
- Decommission Device
- Device Status Check
- On-Demand Read
- Remote Connect
- Remote Disconnect

However, most head-end systems provide additional commands beyond these. This section outlines the steps involved in creating custom commands, including:

- **Create Command (Activity) Business Objects**
- **Create Processing Roles and Configure Processing Methods**
- **Create Outbound and Inbound Communications**
- **Configure Middleware to Support New Messages**

Create Command (Activity) Business Objects

Each command available through Oracle Utilities Smart Grid Gateway is defined as an activity business object. For each custom command you wish to create, you must create an activity business object. The specifics of each business object are based on the type of command to be issuesd to the head-end system. The table below provides an example of the “Remote Connect” activity business object used by the Oracle Utilities Smart Grid Gateway.

Option	Description
Business Object	D1-RemoteConnect
Description	Remote Connect
Maintenance Object	D1-ACTIVITY (Activity)
Application Service	D1-REMOTECONNECTBOAS (Remote Connect BO)
Instance Control	Allow New Instances
Options	• Applicable Processing Role: D1RC (Remote Connect) • Applicable Processing Role: D1SC (On-Demand Read - Scalar) • Applicable Processing Role: D1LC (Load Check) • Related Administration BO: D1-RemoteConnectType (Remote Connect Type) • Display UI Map: D1-RemoteConnectDisplay (Remote Connect - Display) • Portal Navigation Option: d1ActivityNavOpt (Activity Navigation Option) • Display Map Service Script: D1-RtRCDtls (Remote Connect - Retreive Details for Display) • Maintenance UI Map: D1-RemoteConnectMaint (Remote Connect - Maintenance)

Option	Description
Algorithms	• Information: D1-CRAINFO (Command Request Activity Information) • Pre-Processing: D1-DETACTTYP (Determine Activity Type) • Pre-Processing: D1-DMRO (Default Measurement Requested Override) • Pre-Processing: D1-DDR (Determine Device and Recipient) • Validation: D1GINPVAL (Common Input Validation • Validation: D1-VALMDEST (Validate Measurement Destination) • Validation: D1-VALMREQO (Validate Measurement Requested Override)

Lifecycle
 	
Note: Does not include "Transition to Default Next Status" monitor algorithms.
	• Pending
	• Validation
	    • Enter: D1-VALACTTDI (Validate Actvity Type and Transition to Error State If Invalid)
	    • Enter: D1-VALDVCNCD (Validate Device is not already Connected)
	    • Enter: D1-VHCPODR (Validate Head-end's Capability to perform On-Demand Read)
	    • Enter: D1-VHCPCD (Validate Head-end's Capability to perform Connect Disconnect)
	    • Enter: D1-CECD (Check for existing future disconnects)
	• Validation Error
	    • Monitor: D1-WTTMOUT (Wait Time Out)
	    • Monitor: D1-RBOE (Rety BO in Error)
	    • Enter: D1-CTDEBOE (Create To Do Entry for BO in Error)
	    • Exit: D1-GTDCBO (Generic To Do Completion for BOs)
	• Waiting for Effective Date
	    • Monitor: D1-WAITEFFDT (Wait for Effective Date)
	    • Enter: D1-RRER (Send Received Response to External Requester)
	• Connection Ready
	    • Enter: D1-CLCOC (Create Load Check Outbound Communication)
	    • Enter: D1-CODROC (Create On-Demand Read for Start Measurement)
	    • Enter: D1-CRCOC (Create Remote Connect Outbound Communication)

Option	Description

Lifecycle (continued)
 	
Note: Does not include “Transition to Default Next Status” monitor algorithms.
	• Communication in Progress
	• Communication Error
	    • Same as Validation Error (above)
	• Retry
	    • Enter: D1-COOC (Cancel Outstanding Outbound Communication)
	• Execute Completion Events
	    • Enter: D1-EXCMPEVTS (Execute Completion Events)
	• Completion Events Error
	    • Same as Validation Error (above)
	• Waiting for Measurement
	    • Monitor: D1-RMVCE (Retrieve Measurements via Completion Events)
	    • Monitor: D1-RFINSC (Retrieve Scalar Final Measurements)
	    • Monitor: D1-RSINIMS (Retrieve Scalar Initial Measurements)
	    • Monitor: D1-WFMTO (Wait for Measurement Time Out)
	• Wait Expired
	    • Same as Validation Error (above)
	• Completed
	    • Enter: D1-SRCNTEA (Send Remote Connect Notification to Edge Application)
	    • Enter: D1-SODRTEA (Send Start Measurement to Edge Application)
	• Discarded
	    • Enter: D1-COOC (Cancel Outstanding Outbound Communication)
	    • Enter: D1-COCE (Cancel Outstanding Completion Events)
	    • Enter: D1-FAILPA (Fail Parent Activity)
	    • Enter: D1-FRER (Sesnd Fail Response to External Requester)

Create Processing Roles and Configure Processing Methods

After the command activity business object has been created, you must also create a processing role for the command, and associate to your activity business object and the appropriate processing method business object, and then add the processing role/method for the command to the service provider that represents the head-end system.

Create Processing Role For The Command

To create a processing role for your new command, add a value that designates your command to the PROC_ROLE_FLG lookup.

For example, suppose you were creating a custom command to support retrieving outage events for a device. Your lookup value might look like this:

Lookup Field Value	Description (Java Value Name)
CMRO	Retrieve Outage Events (retreiveOutageEvents)

Refer to the Oracle Utilities Application Framework documentation for more information about creating values for lookup fields.

Add Processing Role to Processing Method Business Object

Once the processing role lookup has been created, you have to add it as an Applicable Processing Role on the processing method business object that will be used to determine how to send out outbound communications. For most Oracle Utilities Smart Grid Gateway commands, this is the D1-HowToCreateActivityOBComm (How To Create OB Communication/Send OB Message) business object.

To continue the example of the “Retreive Outage Events” command, you would add the following option to the D1-HowToCreateActivityOBComm business object on the Main tab of the Business Object portal:

Option Type	Option Value
Applicable Processing Role	CMRO

Add Processing Role to Command Business Object

You must also add your processing role lookup as an Initiate Command Processing Role on the command (activity) business object you just created.

To continue the example of the “Retreive Outage Events” command, you would add the following option to the your activity business object on the Main tab of the Business Object portal:

Option Type	Option Value
Initiate Command Processing Role	CMRO

Add Processing Role/Method to Service Provider

You must also add your new processing role and corresponding processing method to the service provider that represents the head-end system.

To continue the example of the “Retreive Outage Events” command, you would add the following processing method to your head-end system service provider.

Processing Role	Processing Method
Retrieve Outage Events (CMRO)	D1-HowToCreateActivityOBComm (How To Create OB Communication/Send OB Message) This processing method specifies the type of outbound communication business object and outbound message type to use when sending outbound communications to a head-end system as a result of a retreive outage events command.

Create Outbound and Inbound Communications

The next step in enabling your new command is to create any necessary outbound and inbound communications to allow the messages related to the command to be sent/received to and from the head-end system. See **Adapters for Issuing and Initiating Commands** on page 9-16 for more information about creating outbound and inbound communications.

Configure Middleware to Support New Messages

Lastly, you need to configure your middleware to support transformation of the communications used by your new command. For example, if your middleware uses XSLT to perform transformation, you need to add specifics about how to transform messages related to “Retreive Outage Events” command.

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>
```

```
        <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
<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.

Element	Description
<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: • D1IL (Initial Load) • D1MO (Manual) • D1ES (Estimation) Defaults to D1IL (Initial Load) if not supplied.
<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
</externalTimeZone>	External Time Zone
<spi>	Seconds-per-interval
<ccond>	Measurement condition

Element	Description
<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)
<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: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: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" />
                                <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" />
                              <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: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: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_HeadEn
d_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: • 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 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:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="version" type="xsd:decimal" minOccurs="0" />
        <xsd:element name="deviceIdentifierNumber" 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: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" />
```

```
                                <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: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 form 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 received 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 $SPLEBASE/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, ForegnDestinationQueueConnectionFactory-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 9-47). 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 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.

Integrating with an Outage Management System

This section provides an overview of how Oracle Utilities Smart Grid Gateway supports integration with an outage management system such as Oracle Utilities Network Management System. It includes the following topics:

- **Integrating with Oracle Utilities Network Management System**
- **Base Package Integration Business Objects**
- **XAI Inbound Services**
- **Master Configurations**

Integrating with Oracle Utilities Network Management System

An integration of Oracle Utilities Smart Grid Gateway with Oracle Utilities Network Management System provides power outage information, restoration information, and message filtering in support of the Outage Management System capability of Oracle Utilities Network Management System. It assists in outage detection, outage verification, and restoration verification. The integration supports the following functionality:

- Meter status information is provided to Oracle Utilities Network Management System.
- "Meter ping" requests are received from Oracle Utilities Network Management System for sending a device status check to the AMI head-end for a power-on status from one or more meters. Ping responses are then sent back to Oracle Utilities Network Management System. This functionality uses the device status check command and bulk command processing in Oracle Utilities Smart Grid Gateway. For more information about bulk command processing, see **Understanding Bulk Commands** on page 7-13.

  Note: The meter ping functionality in this integration is supported only for adapters that support the Device Status Check/Ping command.
- Device event notifications from Oracle Utilities Smart Grid Gateway to Oracle Utilities Network Management System can be suppressed for one or more meters. By suppressing notifications you can prevent messages from being sent from bad meters that provide false statuses, or meters that are being put into a test mode.

This integration process is supported through a set of business objects, XAI inbound communications, and a master configuration. Refer to the *Oracle Utilities Smart Grid Gateway Integration for Outage Operations Implementation Guide* for more information about this integration and the communication processes used by this integration.

Base Package Integration Business Objects

The table below lists the base package integration business objects used by Oracle Utilities Smart Grid Gateway.

Business Objects	Description
D1-DeviceEventNotification	This outbound message business object defines the fields necessary for notifying a subscribing application of a device event received by MDM or SGG. It contains the core device event fields, the device event type code and description, and the standard event name code and description.

Business Objects	Description
D1-DeviceEventNotifSuppType	This activity type business object defines which device events will be suppressed by a suppression of this type. You can list device event categories to be suppressed, and for each device event category you can list those device event types that should be excluded from the suppression. As a default, any device event with that category will be considered suppressed unless it is explicitly excluded.
D1-DeviceEventNotifSuppress	This activity business object represents a suppression of device event notifications to a particular service provider (also called an external system or a subscriber) for a particular service point. The suppression can be initiated by providing either service point information, device information, or both. While a suppression is active, any device events defined on the suppression type that are received for the associated service point will not be sent to the service provider. To restore the sending of device event notifications of device events of that type the suppression would need to be deactivated by transitioning it to the unsuppressed status. A suppression only applies to a single service provider.
D1-OutboundMessageNotifSupp	This outbound message business object defines the fields necessary for notifying a service provider (also called an external system or a subscriber) of a suppression or the removal of a suppression. The message may contain one to many service points.
D1-SuppressionLite	This business object is used to update the send suppression notification flag of the suppression activities.

XAI Inbound Services

The table below lists the pre-configured XAI Inbound Service that is used to process notification suppression/unsuppression requests from Oracle Utilities Network Management System.

XAI Inbound Service	Description	Schema Name
D1-MaintainSuppression	Maintain Suppression XAI Inbound	D1-MAINTSUPP

Master Configurations

The table below lists the master configuration used to support device event notification suppression.

Master Configuration	Description
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 the system 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.

Chapter 10

Configuration Examples and Sample Implementation

This chapter provides configuration examples and a simple example implementation of Oracle Utilities Smart Grid Gateway.

- **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	• **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	• **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	• **Time of Use**: ON • **Description**: On Peak • **Color**: #CCFFCC • **Priority**: 1
Time of Use: Off Peak Business Object: D1-TimeOfUse	• **Time of Use**: OFF • **Description**: Off Peak • **Color**: #008000 • **Priority**: 2

TOU Group
Business Object: D1-TOUGroup
	• **TOU Group**: ON-OFF
	• **Description**: On / Off
	• **TOU Section** (Priority / TOU):
	  • 1 / On Peak
	  • 2 / Off Peak

TOU Map Template
Business Object: D2-TOUMapTemplate
	• **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**:
	  • **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	• **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	• **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	• **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	• **Description**: AM Critical Peak Period • **Dynamic Option Type**: AM Critical Peak Period • **Status**: Active
Dynamic Option Event Business Object: D2-DynamicOptionEvent	• **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	• **Service Quantity Identifier**: CONSUMED_KWH • **Description**: Consumed KWH • **Decimal Positions**: 2
Service Quantity Identifier: Generated KWH Business Object: D1-ServiceQuantityIdentifer	• **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	• **External System**: METERSUS • **Description**: Meters-R-Us
Service Provider: Head-End System Business Object: D1-HeadEndSystem	• **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	• **External System**: CCB • **Description**: Customer Care and Billing
Service Provider: External Application Business Object: D1-ExternalApplication	• **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
	• **Service Provider**: Meters-R-Us
	• **Processing Role**: Initial Measurement Creation
	• **Description**: Initial Measurement Creation
	• **Status**: Active
	• **Default Processing Method** (Business Object): D1-IntnialLoadIMDInterval
	• **Override Processing Method**:
	    • **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	• **Standard Event Name**: DeviceTamperAlert • **Description**: Tamper Alert • **Usage Flag**: Active
Device Event Mapping Extendable Lookup Business Object: D1-DeviceEventMappingLookup	• **Head-End System Event Name**: Tamper Alarm • **Description**: Tamper Alarm • **Standard Event Name**: Tamper Alert
Processing Method: Initial Measurement Creation Business Object: D1-HowToCreateMCInformation	• **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	• **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	• **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	• **Device Configuration Type**: KWH-INTERVAL • **Description**: KWH - Interval Channel • **Device Configuration Business Object**: Device Configuration • **Service Type**: Electric • **Valid Measuring Component Types**: • KWH - 60 Minutes

Device Type Business Object: D1-SmartMeterType	• **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**:
	  • KWH - Interval Channel
	• **Valid Head-End Systems**:
	  • Meters-R-Us

Object	Data and Field Values
Device Business Object: D1-SmartMeter	• **Device Type**: Electric Smart Meter - Interval KWH • **Serial Number**: 12345 • **Head-End System**: Meters-R-Us • **Status**: Active
Device Configuration Business Object: D1-DeviceConfiguration	• **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	• **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	• **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	• **Device Configuration Type**: KWH-SCALAR • **Description**: KWH - Scalar • **Device Configuration Business Object**: Device Configuration • **Service Type**: Electric • **Valid Measuring Component Types**: • Auto Read - Electric - KWH

Device Type
Business Object: D1-SmartMeterType
	• **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**:
	  • KWH - Scalar
	• **Valid Head-End Systems**:
	  • Meters-R-Us

Device Business Object: D1-SmartMeter	• **Device Type**: Electric Smart Meter - Scalar KWH • **Serial Number**: 54321 • **Head-End System**: Meters-R-Us • **Status**: Active
Device Configuration Business Object: D1-DeviceConfiguration	• **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	• **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	• **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	• **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	• **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**:
	    • KWH - 60 Minutes
	    • KWH - Auto Read

Object	Data and Field Values

Device Type
Business Object: D1-SmartMeterType
	• **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**:
	    • KWH - Two Channel (Interval and Scalar)
	• **Valid Head-End Systems**:
	    • Meters-R-Us

Device Business Object: D1-SmartMeter	• **Device Type**: Electric Smart Meter - KWH • **Serial Number**: 12345 • **Head-End System**: Meters-R-Us • **Status**: Active
Device Configuration Business Object: D1-DeviceConfiguration	• **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	• **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	• **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	• **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	• **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	• **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	• **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	• **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	• **Device Configuration Type**: KWH-MANUAL • **Description**: KWH - Manual • **Device Configuration Business Object**: Device Configuration • **Service Type**: Electric • **Valid Measuring Component Types**: • Electric KWH
Device Type Business Object: D1-ManualMeterType	• **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**: • KWH - Manual
Device Business Object: D1-ManualMeter	• **Device Type**: Manual Meter - Scalar KWH • **Serial Number**: 54321 • **Status**: Active
Device Configuration Business Object: D1-DeviceConfiguration	• **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	• **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	• **Item Configuration Type**: KWH-LAMP • **Description**: KWH - Street Lamp • **Item Configuration Business Object**: Device Configuration • **Service Type**: Electric
Device Type Business Object: D1-ItemType	• **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**: • KWH - Street Lamp
Device Type Service Quantity	• **Item Type**: Street Lamp - KWH • **Effective Date**: 01-01-2014 • **Device Type Service Quantity**: 15
Device Business Object: D1-Item	• **Item Type**: Street Lamp - KWH • **Status**: Active • **Serial Number**: 54321

Object	Data and Field Values
Device Configuration Business Object: D1-DeviceConfiguration	• **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	• **Item Configuration Type**: KWH-LAMP • **Description**: KWH - Street Lamp • **Item Configuration Business Object**: Device Configuration • **Service Type**: Electric
Device Type Business Object: D1-ItemType	• **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**: • KWH - Street Lamp
Device Type Service Quantity	• **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	• **Description**: Transformer 1401 - 50KVA • **Facility Type**: Transformer • **Service Type**: Electric • **Status**: Active • **External ID**: 12345
Network Location Business Object: D1-ElectricityNetworkLocation	• **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	• **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	• **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	• **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	• **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**: • **Device Type**: Manual Meter - Scalar KWH

Object	Data and Field Values

Service Point
Business Object: D1-ServicePoint
	• **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**:
	  • **Measurement Cycle**: Electric Residential - Monthly
	  • **Route**: Route 1
	  • **Sequence**: 1

Install Event
Business Object: D1-InstallEvent
	• **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**:
	  • **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	• **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**: • **Device Type**: Street Lamp - KWH
Service Point Business Object: D1-ServicePoint	• **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
	• **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**:
	    • **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	• **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**: • **Device Type**: Street Lamp - KWH

Service Point
Business Object: D1-ServicePoint
	• **Service Point Type**: Street Light - Multiple
	• **Status**: Active
	• **Address**: 27 Park Street, Wakefield, MA 99999, US
	• **External Service Point ID**: 12345
	• **Multi-Item Information**:
	  • **Start Date/Time**: 01-01-2014 12:00:00AM
	  • **Installation Override Quantity (Daily)**: 12
	  • **Multi-Item Count**:
	    • **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	• **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	• **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	• **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)

<table>
<tr><th>Object</th><th>Data and Field Values</th></tr>
<tr>
<td>Service Issue Monitor Type Business Object: D1-ServiceIssueMonitorType

Service Task Business Object: Service Issue Monitor</td>
<td>

Service Task Type: TAMPERING
Description: Tamper
Service Task Class: Service Issue Monitor
Approval Required: Not Required
Evaluation Criteria: (broken seal)

Sequence: 10
Evaluation Criteria Relationship: OR
Service Issue Monitor Evaluation Types: Device Event
Evaluation Details:

Device Event Type: Broken Seal

Number of Occurrences: 1
Number of Days Back: 1

Evaluation Criteria: (tamper)

Sequence: 20
Evaluation Criteria Relationship: OR
Service Issue Monitor Evaluation Types: Device Event
Evaluation Details:

Device Event Type: Tamper

Number of Occurrences: 1
Number of Days Back: 1

Service Investigative Order:

Service Investigative Order Activity Type: Field Activity Type (custom)
Field Task Type:

Service Issue Monitor Discard Rules:

If Activity Found - Different Monitor Type: No
If Completed Activity Found: No

</td>
</tr>
</table>

Sample Implementation

This section describes the steps involved in configuring Oracle Utilities Smart Grid Gateway 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.

Note: References to objects and options pre-fixed with “D2” are available only with Oracle Utilities Meter Data Management, and included for illustrative purposes only.

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	• Residential (approximately 50,000) • Commercial (approximately 1,000) • Industrial (approximately 100)
Types of Meters	• Residential: Scalar, single register (manual) • Commercial: Interval channel/scalar register (smart) • Industrial: Two interval channels (smart)
Meter Manufacturers / Head-End Systems	• MetersRUs: Scalar (used for residential customers) • MeterTech, Inc.: Interval (used for commercial and industrial customer)
Usage Metered	• Residential: KWH (scalar) • Commercial: KWH (interval and scalar) • Industrial: KWH and KVARH (interval)
Readings/Measurement Data	• Scalar (for residential and commercial) • Interval (for commercial and industrial, 15 minutes) Both head-end systems use their own UOM codes.
Device Events	Both head-end systems listed above send standard and “paired” device events, and use their own device event names.

Requirement	Description
Meter Commands	Both head-end systems support the following commands: • Commission Device • Decommission Device • Device Status Check • On-Demand Read (interval and scalar) • Remote Connect • Remote Disconnect Each command sends and received a single message when invoking the command.
Validation, Editing, and Estimation Rules	Validation, editing, and estimation is performed by Oracle Utilities Meter Data Management.*
Bill Determinants	Bill determinant calculations are performed by Oracle Utilities Meter Data Management.*
Billing System	Oracle Utilities Customer Care and Billing

*See **Chapter 16**: **Configuration Examples and Sample Implementation** in the Oracle Utilities Meter Data Management Configuration Guide for examples of implementing VEE and bill determinant calculations.

Implementation Steps

This section outlines the steps in configuring Oracle Utilities Smart Grid Gateway 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

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)
  - Valid Command Request BO: D1-DeviceCommission, etc.* (see below)
- Commercial: CM-IntChanScalarReg
  - Install Event BO: D1-SmartMeterInstallEvent(see above)
  - Valid Command Request BO: D1-DeviceCommission, etc.* (see below)
- Industrial: CM-Interval2Channels
  - Install Event BO: D1-SmartMeterInstallEvent (see above)
  - Valid Command Request BO: D1-DeviceCommission, etc.* (see below)

***Note**: Each device business object should specify the specific commands available for devices based on that business object

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

Measurements: A single measurement business object for all final measurements, as follows:

- Final Measurement: CM-FinalMeasurement

UOM Mapping: UOM mapping extendable lookup business objects for each head-end system, as follows:

- MetersRUs: CM-MRUUOMMapping
- MeterTech, Inc.: CM-MTUOMMapping

Device Events

For device events, we would need to create the following:

Device Events: This implementation can use the base package device event business objects for "standard" and "paired" device events, as follows:

- Standard Device Event: D1-StandardDeviceEvent
- Paired Device Event - First: D1-PairedDeviceEventFirstDeviceEvent
- Paired Device Event - Last: D1-PairedDeviceEventLastDeviceEvent

Device Event Mapping: Device event mapping extendable lookup business objects for each head-end system, as follows:

- MetersRUs: CM-MRUDeviceEventMapping
- MeterTech, Inc.: CM-MTDeviceEventMapping

Meter Commands

For meter commands data, we would need to create the following:

Activities: For activities, this implementation can use the base package business objects, including the activity command business objects.

Outbound Communication: Outbound communication business objects for each command, as follows:

- Commission Device: CM-InitiateCommissionDevice
- Decommission Device: CM-InitiateDecommissionDevice
- Device Status Check: CM-InitiateDeviceStatusCheck
- On-Demand Read Interval: CM-InitiateOnDemandReadInterval
- On-Demand Read Scalar: CM-InitiateOnDemandReadScalar
- Remote Connect: CM-InitiateRemoteConnect
- Remote Disconnect: CM-InitiateRemoteDisconnect

Inbound Communication: Inbound communication business objects for each command response, as follows:

- Commission Device: CM-CommissionDeviceResponse
- Decommission Device: CM-DecommissionDeviceResponse
- Device Status Check: Cm-DeviceStatusCheckResponse
- On-Demand Read Interval: CM-OnDemandReadIntervalResponse

- On-Demand Read Scalar: CM-OnDemandReadScalarResponse
- Remote Connect: CM-RemoteConnectResponse
- Remote Disconnect: CM-RemoteDisconnectResponse

Completion Events: This implementation can use the base package completion event business objects.

Note: Because this implementation supports two head-end systems, you would need business objects for each communication for each head-end system. This list is for only one head-end system.

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)

Communication Type	One for each type of communication (outbound or inbound) as follows: • Commission Device / Response Business Object (Initiate): CM-InitiateCommissionDeviceType Business Object (Response): CM-CommissionDeviceResponseType • Decommission Device / Response Business Object (Initiate): CM-InitiateDecommissionDeviceType Business Object (Response): CM-DecommissionDeviceResponseType • Device Status Check / Response Business Object (Initiate): CM-InitiateDeviceStatusCheckType Business Object (Response): CM-DeviceStatusCheckResponseType • On-Demand Read Interval / Response Business Object (Initiate): CM-InitiateOnDemandReadIntervalType Business Object (Response): CM-OnDemandReadIntervalResponseType • On-Demand Read Scalar / Response Business Object (Initiate): CM-InitiateOnDemandReadScalarType Business Object (Response): CM-OnDemandReadScalarResponseType • Remote Connect / Response Business Object (Initiate): CM-InitiateRemoteConnectType Business Object (Response): CM-RemoteConnectResponseType • Remote Disconnect / Response Business Object (Initiate): CM-InitiateRemoteDisconnectType Business Object (Response): CM-RemoteDisconnectResponseType Note: The communication business objects described under “Create Business Objects” serve as the Related Transaction BO for each of the above.
Contact Type	One for each type of customer, as follows: • RESIDENTIAL (Residential - Person) Business Object: CM-ResPerson • COMMERCIAL (Commercial - Business) Business Object: CM-ComBusiness • INDUSTRIAL (Industrial - Business) Business Object: CM-IndBusiness

Device Event Type	One for each type of device event sent from the head-end systems. Examples might include: • BROKEN SEAL Business Object: D1-StandardDeviceEvent • OUTAGE Business Object: D1-PairedDeviceEventFirstDeviceEvent • POWER RESTORATION Business Object: D1-PairedDeviceEventLastDeviceEvent • TAMPER Business Object: D1-StandardDeviceEvent
Exception Type	One for each VEE rule, as appropriate.
Factor	N/A
Market	Single market: • SMALLTOWNUSA
Measurement Cycle	Twenty cycles for each type of customer, as follows: • RESMC01, RESMC02, ..., RESMC20 • COMMC01, COMMC02, ..., COMMC20 • INDMC01, INDMC02, ..., INDMC20

Measurement Cycle Schedule	One per measurement cycle per month, as follows:
	• RESMC01:
	  • Scheduled Selection Date: 08/02/2010
	  • Expected Work Date: 08/03/2010
	• RESMC02:
	  • Scheduled Selection Date: 08/03/2010
	  • Expected Work Date: 08/04/2010
	• RESMC03:
	  • ...

Service Provider	One for each head-end system, and one for the billing system, as follows: • 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 * See **Service Providers** on page 10-33 for additional details.
Service Quantity Identifier	N/A
Service Type	Single service type: • ELECTRIC (Electric)

Manufacturer	One for each manufacturer, as follows:
	• METERSRUS (Meters R Us)
	    • Models: RES2010
	• METERTECH (MeterTech, Inc.)
	    • Models: COM2010, IND2010

Unit of Measure	One for each type of metered usage and for calculated usage, as follows: • KVA (calculated) • KVAR (used in KVA calculation) • KVARH (measured) • KW (derived) • KWH (measured) * All UOMs use the ELECTRIC service type.
Measuring Component Type	Five measuring component types, as follows: • RESSCALAR (Residential Scalar Register) Business Object: CM-ResScalarRegister • COMINTERVAL (Commercial Interval Channel) Business Object: CM-IntervalChannel • COMSCALAR (Commercial Scalar Register) Business Object: CM-ScalarValRegister • INDINTERVALKVARH (Industrial KVARH Interval Channel) Business Object: CM-IntervalChannel • INDINTERVALKWH (Industrial KWH Interval Channel) Business Object: CM-IntervalChannel * All measuring component types use the ELECTRIC service type. See **Measuring Component Types** on page 10-36 for additional details.

Device Configuration Type	One device configuration type for each type of meter installed, as follows:
	• RESDVCCFG (Residential Device Configuration)
	Valid Measuring Component Types:
	    • RESSCALAR (Residential Scalar Register)
	• COMDVCCFG (Commercial Device Configuration)
	Valid Measuring Component Types:
	    • COMSCALAR (Commercial Scalar Register)
	    • COMINTERVAL (Commercial Interval Channel)
	• INDDVCCFG (Industrial Device Configuration)
	Valid Measuring Component Types:
	    • INDINTERVALKVARH (Industrial KVARH Interval Channel)
	    • INDINTERVALKWH (Industrial KWH Interval Channel)
	*All device configuration types use the ELECTRIC service type.

Device Type	One device type for each type of meter installed, as follows: • RESDVC (Residential Device) Business Object: CM-ScalarRegister Valid Device Configuration Types: • RESDVCCFG (Residential Device Configurations) • COMDVC (Commercial Device) Business Object: CM-IntChanScalarReg Valid Device Configuration Types: • COMDVCCFG (Commercial Device Configurations) • INDDVC (Industrial Device) Business Object: CM-Interval2Channels Valid Device Configuration Types: • INDDVCCFG (Industrial Device Configurations) *All device types use the ELECTRIC service type.
Service Point Type	One for each type of customer rule, as follows: • RESIDENTIAL (Residential) Business Object: CM-ResidentialSP Valid Device Types: • RESDVC (Residential Devices) • COMMERCIAL (Commercial) Business Object: CM-CommercialSP Valid Device Types: • COMDVC (Commercial Devices) • INDUSTRIAL (Industrial) Business Object: CM-IndustrialSP Valid Device Types: • INDDVC (Industrial Devices) * All service point types use ELECTRIC service type

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:

Processing Role	Processing Method	Default Processing Method	Override Processing Methods
Initial Measurement Creation	How To Create MC Related Information	CM-InitialLoadIMDScalar (Business Object)	MC Type: Residential Scalar Register (RESSCALAR) Business Object: CM-InitialLoadIMDScalar
Device Commission	How to Create OB Communication / Send OB Message	CM-InitiateCommissionDevice (Business Object)	
Device Decomission	How to Create OB Communication / Send OB Message	CM-InitiateDecommissionDevice (Business Object)	
Device Event Mapping	How to Process Device Related Information	CM-MRSDeviceEventMapping (Business Object)	
Device Status Check	How to Create OB Communication / Send OB Message	CM-InitiateDeviceStatusCheck (Business Object)	
On-Demand Read (Interval)	How to Create OB Communication / Send OB Message	CM-InitiateOnDemandReadInterval (Business Object)	
On-Demand Read (Scalar)	How to Create OB Communication / Send OB Message	CM-InitiateOnDemandReadScalar (Business Object)	
Remote Connect	How to Create OB Communication / Send OB Message	CM-InitiateRemoteConnect (Business Object)	
Remote Disconnect	How to Create OB Communication / Send OB Message	CM-InitiateRemoteDisconnect (Business Object)	
UOM Mapping	How to Process Device Related Information	CM-MRUUOMMapping (Business Object)	

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:

Processing Role	Processing Method	Default Processing Method	Override Processing Methods
Initial Measurement Creation	How To Create MC Related Information	CM-InitialLoadIMDInterval (Business Object)	MC Type: Residential Scalar Register (RESSCALAR) / Business Object: CM-InitialLoadIMDScalar MC Type: Commercial Scalar Register (COMSCALAR) / Business Object: CM-InitialLoadIMDScalar
Device Commission	How to Create OB Communication / Send OB Message	CM-InitiateCommissionDevice (Business Object)	
Device Decomission	How to Create OB Communication / Send OB Message	CM-InitiateDecommissionDevice (Business Object)	
Device Event Mapping	How to Process Device Related Information	CM-MTDeviceEventMapping (Business Object)	
Device Status Check	How to Create OB Communication / Send OB Message	CM-InitiateDeviceStatusCheck (Business Object)	
On-Demand Read (Interval)	How to Create OB Communication / Send OB Message	CM-InitiateOnDemandReadInterval (Business Object)	
On-Demand Read (Scalar)	How to Create OB Communication / Send OB Message	CM-InitiateOnDemandReadScalar (Business Object)	
Remote Connect	How to Create OB Communication / Send OB Message	CM-InitiateRemoteConnect (Business Object)	
Remote Disconnect	How to Create OB Communication / Send OB Message	CM-InitiateRemoteDisconnect (Business Object)	
UOM Mapping	How to Process Device Related Information	CM-MTUOMMapping (Business Object)	

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:

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)

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

Other Components

Note that for each communication you create, you must also create corresponding components of the following types used to send/received messages from the middleware. These include:

Outbound Communications

- Outbound Message Type
- XAI Sender (XAI)
- External System

Inbound Communications

- XAI Inbound Service

See **Chapter 9**: **Integrating Oracle Utilities Smart Grid Gateway with Other Systems** for more information about creating these components when configuring communications with head-end systems and middleware.

Chapter 11

Smart Grid Gateway Adapters

This chapter describes the components provided with Oracle Utilities Smart Grid Gateway adapters. This includes:

- **Smart Grid Gateway Adapter Overview**
- **Oracle Service Bus Projects**
- **Adapter-Specific Business Objects**
- **Oracle Business Process Execution Language Composites**
- **SGG Adapter Configuration Portal**

Adapter-specific configuration guides provide additional details for each of the productized adapters.

Smart Grid Gateway Adapter Overview

This section provides a high-level overview of the components provided with Smart Grid Gateway adapters, including:

- **Oracle Service Bus Projects**
- **Oracle Utilities Service and Measurement Data Foundation**
- **Adapter-Specific Business Objects**
- **Oracle Business Process Execution Language Composites**
- **Available Smart Grid Gateway Adapters**

Oracle Service Bus Projects

Smart Grid Gateway adapters use Oracle Service Bus (OSB) to provide import/upload functionality for usage readings and device events. Adapters include both base package and customizable projects, to allow implementations to customize their OSB implementation as needed to meet specific requirements.

See **Oracle Service Bus Projects** on page 11-4 for more information about OSB projects used by Smart Grid Gateway adapters.

Oracle Utilities Service and Measurement Data Foundation

Smart Grid Gateway adapters are installed along with the Oracle Utilities Service and Measurement Data Foundation (D1), which contains business objects and other configurations used by all adapters. For example, business objects for each of the smart meter commands supported by Smart Gird Gateway are defined in the Meter Data Framework. These business objects create vendor-specific communications based on configuration of the head-end system service provider.

The components in the Service and Measurement Data Foundation used by Smart Grid Gateway are described in the previous chapters of this book.

Adapter-Specific Business Objects

Smart Grid Gateway adapters include adapter-specific business objects for the specific outbound and inbound communications used to execute and respond to smart meter commands. Adapters also make use of extendable lookup business objects to allow implementations to define values for different types of data (data mapping, etc.).

See **Adapter-Specific Business Objects** on page 11-12 for more information about business objects used by Smart Grid Gateway adapters.

Oracle Business Process Execution Language Composites

Smart Grid Gateway adapters use Oracle Business Process Execution Language (BPEL) composites to send and receive messages between Smart Grid Gateway and vendor-specific head-end systems. This communication process includes transforming messages from the Smart Grid Gateway for to the vendor’s format (and the reverse when receiving messages).

See **Oracle Business Process Execution Language Composites** on page 11-15 for more information about BPEL composites used by Smart Grid Gateway adapters.

Available Smart Grid Gateway Adapters

The vendor-based adapters available for Smart Grid Gateway include the following:

Adapter	Currently Supported Head-End System
Adapter for Landis+Gyr (D3)	Gridstream Command Center 6.3
Adapter for Echelon (D4)	Echelon NES 5.2
MV-90 Adapter for Itron (D5)	Supports the .mv9 binary, mainframe data format
Adapter for Sensus (D6)	Sensus RNI 3.1
Adapter for Silver Springs Network (D7)	UtilityIQ Version 4.10
Adapter for Itron OpenWay (D8)	6.1
Adapter Development Kit (DG)	MultiSpeak 4.1

Refer to individual Adapter Configuration Guides for more information about these adapters.

Oracle Service Bus Projects

Oracle Utilities Smart Grid Gateway adapters use Oracle Service Bus (OSB) projects to provide upload functionality for usage and device events. This section outlines the types of OSB projects provided with Smart Grid Gateway adapters, and describes the components of each type of OSB project. This includes:

- **Understanding the Upload Process**
- **Adapter OSB Projects**
- **Usage Upload - BASE**
- **Usage Upload - CM**
- **Event Upload - BASE**
- **Event Upload - CM**
- **Working with Adapter OSB Projects**

> **Note**: This document describes the components of the OSB projects in terms of how they are used by Smart Grid Gateway adapters. Refer to Oracle Service Bus documentation for more information about OSB projects and the types of components provided in the projects described here.

Understanding the Upload Process

This section provides a high-level description of the upload process used for both usage and device events., including:

- **Upload Process**
- **OSB's Roles in the Upload Process**

Upload Process

When usage and/or device events are sent to a Smart Grid Gateway adapter, the following takes place:

1. The head-end system sends a payload to the adapter.
2. Oracle Service Bus (OSB) receives payload from the head-end system:
   - Parses payload into individual usage readings / device events
   - Transforms individual readers/events into standard (unified) format
   - Invokes seeder for each record
   - Creates upload statistics for each payload
3. The OUAF receives the incoming usage reading and:
   - Determines the device and measuring component for the reading or device event.
   - Maps UOM codes in usage reading to standard UOM codes (readings only)
   - Creates an initial measurement or device event
4. The Service Provider's processing methods define rules for
   - Determining the type of initial measurement or device event to create
   - Mapping vendor-specific UOM codes to standard UOM codes (readings only)
   - Mapping vendor-specific status codes to standard status codes (readings only)

OSB's Roles in the Upload Process

This section provides more details regarding the processing performed by OSB when receiving payloads from head-end systems. This includes the following:

- JCA Adapters monitor a file system for incoming file-based payloads. The polling directory monitored by OSB can be configured based on an implementation's requirements.
- Java objects configured with the OSB projects perform parsing of payloads into individual usage readings or device events.
- Xquery or XSLT scripts are used to perform transformation from head-end system format into a unified format used by all Smart Grid Gateway adapters.
- OSB delivers individual usage or event transactions to JMS queues for processing by SGG adapters.
- OSB creates upload statistics activities for each payload processed. See **Upload Statistics** on page 7-5 for more information about upload statistics activities.

The specific components of each OSB project used to perform the above are outlined in more details in later sections.

Adapter OSB Projects

Smart Grid Gateways include OSB projects based on the upload functionality supported by each adapter. This section provides an overview of the objects used for usage and device event upload.

Usage Upload OSB Projects

Each Smart Grid Gateway adapter provides two OSB projects to support usage upload:

- SGG-Dn-USAGE-BASE
- SGG-Dn-USAGE-CM

> **Note**: The **"Dn"** in the project names designates the specific adapter. For example, "D3" designates the Landis+Gyr adapter, "D6" designates the Sensus adapter, "DG" designates the Adapter Development Kit, etc.

SGG-Dn-USAGE-BASE: Contains base usage upload functionality for transforming incoming vendor-specific usage readings into the standard "unified" format used by Smart Grid Gateway adapters. This project CANNOT be modified by an implementation.

SGG-Dn-USAGE-CM: Contains extendable functionality used when transforming vendor-specific usage readings into the standard "unified" format used by Smart Grid Gateway adapters. This project can be modified to support customizations required by an implementation.

This project also contains variables that an implementation would need to modify. This includes polling directories, JMS queue information, etc.

The "SGG-Dn-USAGE-BASE" project reference services in the "SGG-Dn-USAGE-CM" project It is the "SGG-Dn-USAGE-CM" project that sends data and notifications to Smart Grid Gateway adapters.

Device Event Upload OSB Projects

Each Smart Grid Gateway adapter that supports device event upload provides the following two OSB projects:

- SGG-Dn-EVENT-BASE
- SGG-Dn-EVENT-CM

> **Note**: The **"Dn"** in the project names designates the specific adapter. For example, "D3" designates the Landis+Gyr adapter, "D6" designates the Sensus adapter, "DG" designates the Adapter Development Kit, etc.

SGG-Dn-EVENT-BASE: Contains base device event upload functionality for transforming incoming vendor-specific device events into the standard “unified” format used by Smart Grid Gateway adapters. This project CANNOT be modified by an implementation.

SGG-Dn-EVENT-CM: Contains extendable functionality used when transforming vendor-specific device events into the standard “unified” format used by Smart Grid Gateway adapters. This project can be modified to support customizations required by an implementation.

The “SGG-Dn-EVENT-BASE” projects reference services in the “SGG-Dn-EVENT-CM” projects It is the “SGG-Dn-EVENT-CM” project that sends data and notifications to Smart Grid Gateway adapters.

Usage Upload - BASE

This section outlines the components of “SGG-Dn-USAGE-BASE” OSB projects.

“SGG-Dn-USAGE-BASE” projects define the following:

- JAR files
- JCA Bindings
- Proxy Services
- XML Schemas (IMD format, vendor format, notification format)
- XQueries (used to transform incoming format to “unified” format)

JAR Files

JAR files contain Java objects used by the OSB project to transform and parse payloads.

JCA Bindings

JCA binding resources are used to consume JCA resources when communicating with external systems via OSB. JCA bindings can generate proxy and business services that utilities OB JCA transport to communicate with external systems through JCA adapters.

A JCA file and WSDL are required to create a JCA Transport-based proxy in Oracle Service Bus. These objects are created and delivered with each productized adapter. More information on the JCA File Adapter properties file can be found in the *Oracle Fusion Middleware User's Guide for Technology Adapters*.

The WSDL contained within the JCA Bindings resource is used to describe the JCA File adapter service used for each productized adapter.

Proxy Services

OSB uses proxy services to route messages between business services (such as enterprise Web services and databases) and service clients (such as external applications or other business services).

Typically two proxy services are delivered with each adapter:

- **ProcessingProxyService**: The ProcessingProxyService is a local transport based proxy service. It contains the message flow where the vendor-specific validation and transformation functionalities are implemented. The publishing of processing notifications is implemented in ProcessingproxyService as well.
- **ResultsProcessingProxyService**: The ResultProcessingProxyService contains actions where post-transformation logic will be implemented. Also, it will have an Error Handler responsible only for catching errors occurred after a “Plain” XML is transformed to the result transaction.

XML Schemas

All Smart Grid Gateway adapters include XML schemas that are used to verify the translations from vendor-specific formats to the unified IMD XML format. “SGG-Dn-USAGE-BASE” OSB projects typically contain the following types of XML schemas:

- **InitialLoadIMD Schema**: The InitialLoadIMD schema defines the structure of the measurement data sent from OSB to the OUAF-based components of Smart Grid Gateway.
- **Vendor-specific Schema**: Vendor-specific schemas define the format of the usage data being received from the head-end system. For example, the “LG-Usage-Common” schema defines the format of the usage data being received from the Landis+Gyr head-end system.
- **ProcessNotifications Schema**: Smart Grid Gateway adapters also provide the XML schema that defines the XML used during all notification processing. Notification processing is what drives the creation of upload statistics activities.

XQueries

Smart Grid Gateway adapters leverage Xquery to transform the vendor formatted usage data.

Usage Upload - CM

This section outlines the components of “SGG-Dn-USAGE-CM” OSB projects.

“SGG-Dn-USAGE-CM” projects define the following:

- Business Services
- Proxy Services (including pre-processing and post-processing proxy services)
- Service Accounts
- XQueries (Environment Settings)

Business Services

Business Services control how data is delivered from OSB to the OUAF components of Smart Gird Gateway. For all Smart Grid Gateway adapters, data is transferred via JMS queue. There is a business service for the usage data, and another for the notification data used for upload statistics. JMS information for each type of message is defined within the applicable business service.

Proxy Services

The “InboundProxyService” proxy service for each OSB project contains transport configuration settings. These are stored in the “SG-Dn-USAGE-CM” project so that they can be modified by an implementation.

The transport method for all Smart Grid Gateway adapters is the JCA File Adapter. The JCA Transport Configuration section of the “InboundProxyService” contains polling controls such as polling frequency, polling directory, archive directory and error directory.

In addition, the “External Reference ID” that is stored in this proxy service. This field is compared against the External Reference Id (stored on the service provider) to determine which Service Provider is to be used to process this transaction went sent to the OUAF-based components of the adapter.

Service Accounts

Service accounts provide user names and passwords that proxy services and business services use for outbound authentication or authentication to a local or remote resource, such as an FTP server or a JMS server.

XQueries

XQueries in "SGG-Dn-USAGE-CM" projects include environment settings that allow implementations to customize how usage upload processing is performed by OSB.

Typical environment settings for usage upload include the following

Environment Setting	Description
populateRawIMD	Specifies if the initial measurement data will also store the raw data as received from the AMI vendor.
callPreProcessing	Specifies if the pre-processing proxy service is called (true/false).
callPostProcessing	Specifies if the post-processing proxy service is called (true/false).
destinationRootElementInterval	Holds the name of the XAI Inbound Service for the interval IMD Seeder.
destinationRootElementScalar	Holds the name of the XAI Inbound Service for the scalar IMD Seeder. In most cases it is the same as destinationRootElementInterval.

Specific vendors have unique environment settings. Refer to individual adapter-specific configuration guides for the specific environment settings supported for each adapter.

Event Upload - BASE

This section outlines the components of "SGG-Dn-EVENT-BASE" OSB projects.

"SGG-Dn-EVENT-BASE" projects define the following:

- JAR files
- JCA Bindings
- Proxy Services
- XML Schemas (device event format, vendor format, notification format)
- XQueries (used to transform incoming format to "unified" format)

The components of "SGG-Dn-EVENT-BASE" OSB projects are similar to those used by "SGG-Dn-USAGE-BASE" projects (described above).

XML Schemas

All Smart Grid Gateway adapters include XML schemas that are used to verify the translations from vendor-specific formats to the unified IMD XML format. "SGG-Dn-EVENT-BASE" OSB projects typically contain the following types of XML schemas:

- **DeviceEventSeeder Schema**: The DeviceEventSeeder schema defines the structure of the device event data sent from OSB to the OUAF-based components of Smart Grid Gateway.
- **Vendor-specific Schema**: Vendor-specific schemas define the format of the device event data being received from the head-end system. For example, the "LG-Event" schema defines the format of the device events being received from the Landis+Gyr head-end system.
- **ProcessNotifications Schema**: Smart Grid Gateway adapters also provide the XML schema that defines the XML used during all notification processing. Notification processing is what drives the creation of upload statistics activities.

Event Upload - CM

This section outlines the components of “SGG-Dn-EVENT-CM” OSB projects.

“SGG-Dn-EVENT-CM” projects define the following:

- Business Services
- Proxy Services (including pre-processing and post-processing proxy services)
- Service Accounts
- XQueries (Environment Settings)

The components of “SGG-Dn-EVENT-CM” OSB projects are similar to those used by “SGG-Dn-USAGE-CM” projects (described above).

XQueries

XQueries in “SGG-Dn-EVENT-CM” projects include environment settings that allow implementations to customize how device event upload processing is performed by OSB.

Typical environment settings for device upload include the following

Environment Setting	Description
populateRaw	Specifies if the device event data will also store the raw data as received from the AMI vendor.
callPreProcessing	Specifies if the pre-processing proxy service is called (true/false).
callPostProcessing	Specifies if the post-processing proxy service is called (true/false).
destinationRootElement	Holds the name of the XAI Inbound Service for the device event Seeder.

Specific vendors have unique environment settings. Refer to individual adapter-specific configuration guides for the specific environment settings supported for each adapter.

Working with Adapter OSB Projects

OSB projects provided with Smart Grid Gateway adapters can be accessed using the Oracle Service Bus Console. You access the OSB console by logging on to the OSB server using Internet Explorer. This section outlines some common tasks related to working with OSB projects using the OSB console, including:

- **Connecting to the OSB Console**
- **Changing the Polling Directory**
- **Setting the Minimum Age for File Retrieval**
- **Changing Environment Settings**

Connecting to the OSB Console

The address for the OSB console is:

http://**<OSB_server>**:**<port>**/sbconsole

where:

- **<OSB_server>** is the URL for the OSB server.

- **<port>** is the port in which the OSB server is installed. The default port used with Smart Grid Gateway adapters is 8001.

OSB projects are located in the **Project Explorer**, in the **Operations** menu.

Changing the Polling Directory

The steps below outline how to change the polling directory used with usage upload.

> **Note**: You must have an open Change Session in order to make changes to an OSB project.

1. Log in to the OSB admin console.
2. Select the "SGG-Dn-USAGE-CM" project you wish to modify in the **Project Explorer**.
3. Click **Proxy Services** under the "SGG-Dn-USAGE-CM" project.
4. Select the **InboundProxyService** (Proxy Service).
5. Click **Edit** in the **Change Center**.
6. Click the **Edit** icon in the **JCA Transport Configuration** section.
7. Click the down arrow icon in the **Advanced Settings** bar.
8. Change the "PhysicalDirectory" value as appropriate (to the polling directory you wish to use).
9. Click the **Last >>** button.
10. Click **Save**.

    The PhysicalDirectory now points to the new folder location.
11. Click the **Activate** button in the **Change Center**.

    Enter a note about the change in the **Description** field.
12. Click **Submit**.

Setting the Minimum Age for File Retrieval

The steps below outline how to change the MinimumAge property in the InboundProxyService configuration. Changing this property may be necessary to enable large files to be completely copied to the input directory before they are retrieved for processing. If a file is detected in the input directory and its modification time is within the MinimumAge value of the current time, then the file is not retrieved because it is still potentially being written to.

To change the MinimumAge property, follow these steps:

> **Note**: You must have an open Change Session in order to make changes to an OSB project.

1. Log in to the OSB admin console.
2. Select the "SGG-Dn-USAGE-CM" project you wish to modify in the **Project Explorer**.
3. Click **Proxy Services** under the "SGG-Dn-USAGE-CM" project.
4. Select the **InboundProxyService** (Proxy Service).
5. Click **Edit** in the **Change Center**.
6. Click the **Edit** icon in the **JCA Transport Configuration** section.
7. Click the down arrow icon in the **Advanced Settings** bar.
8. Change the "MinimumAge" value to the amount of time, in seconds, that is required to copy the largest expected file into the input directory.
9. Click the **Last >>** button.

10. Click **Save**.
11. Click the **Activate** button in the **Change Center**.

    Enter a note about the change in the **Description** field.
12. Click **Submit**.
13. Make sure that files copied to the input directory have the current time stamp.

Changing Environment Settings

The steps below outline how to change the environment settings used with event upload.

> **Note**: You must have an open Change Session in order to make changes to an OSB project.

1. Log in to the OSB admin console.
2. Select the “SGG-Dn-EVENT-CM” project you wish to modify in the **Project Explorer**.
3. Click **XQueries** under the “SGG-Dn-USAGE-CM” project.
4. Select the **EnvironmentSettings** (XQuery).
5. Click **Edit**.
6. Edit the values of the environment setting elements as appropriate.

   For example, to disable storing device event data as raw data, you would change the <populateRaw> element as follows:

   ```
   <populateRaw>false</populateRaw>
   ```
7. Click **Save**.

Adapter-Specific Business Objects

This section outlines the types of business objects provided in Smart Grid Gateway adaptesr, including:

- **Initial Measurement Data Business Objects**
- **Outbound Communications Business Objects**
- **Inbound Communications Business Objects**
- **Extendable Lookup Business Objects**

Initial Measurement Data Business Objects

Initial measurement data business objects are used to define vendor-specific initial measurements. All adapters include initial measurement data business objects. These are typically child business objects of the base (D1-owned) initial measurement data business objects (D1-InitialLoadIMDInterval or D1-InitialLoadIMDScalar).

Example: Sensus (D6)

The Sensus adapter includes the following initial measurement data business objects:

Business Object	Description
D6-InitialLoadIMDScalar	Sensus Initial Load IMD - Scalar
D6-InitialLoadIMDInterval	Sensus Initial Load IMD - Interval

Outbound Communications Business Objects

Outbound communications business objects are used to define the outbound messages sent from the Smart Grid Gateway adapters to the head-end systems when executing smart meter commands. All adapters that support two-way communications with a head-end system include outbound communications business objects. There is typically one outbound communication business object for each type of message that can be sent from the adapter to the head-end system. These are most often named based on names used by the head-end system.

Example: Silver Springs Network (D7)

The Silver Springs Network adapter includes the following outbound communications business objects:

Business Object	Description
D7-AddMeterReadJobInterval	SSN - Add Meter Read Job (Interval)
D7-AddMeterReadJobScalar	SSN - Add Meter Read Job (Scalar)
D7-AddPingJob	SSN - Add Ping Job
D7-ConnectDisconnect	SSN - Connect or Disconnect
D7-GetStatus	SSN - Get Status
D7-ReplaceDeviceAtLocForDecomm	SSN - SSN - Replace Device At Location (Decomm)
D7-ReplaceLocation	SSN - Replace Location

Inbound Communications Business Objects

Inbound communications business objects are used to define the inbound messages sent from the head-end system to the Smart Grid Gateway adapters when sending responses based on smart meter commands. All adapters that support two-way communications with a head-end system include inbound communications business objects. There is typically one inbound communication business object for each type of message that can be sent from the head-end system to the adapter. These are most often named based on names used by the head-end system.

Example: Silver Springs Network (D7)

The Silver Springs Network adapter includes the following inbond communications business objects:

Business Object	Description
D7-ConnectDisconnectResp	SSN - Connect or Disconnect Response
D7-GetStatusResponse	SSN - Get Status Response
D7-MeterReadResponseInterval	SSN - Meter Read Response (Interval)
D7-MeterReadResponseScalar	SSN - Meter Read Response (Scalar)
D7-PingJobResponse	SSN - Ping Job Response

Extendable Lookup Business Objects

Extendable lookup business objects are used to define lookup values for certain types of data, including mapping of vendor-specific data to standard equivalents (such as vendor-specific device event names to standard names), defining vendor-specific codes (such as status codes) and others. All adapters include extendable lookup business objects. There are three primary types of extendable lookup business objects included with Smart Grid Gateway adapters:

- **Device Event Mapping**: Used to define how vendor-specific device event names are mapped to standard events names. For example, the same type of event (such as a tampering alert) can have different names in different head-end systems. Device event mapping allows the same types of events received from different head-end systems to use the same standard event name in Smart Grid Gateway., regardless of the head-end system.

  Device event mapping extendable lookup business objects are typically child business objects of the base (D1-owned) device event mapping business object (D1-DeviceEventMappingLookup).

- **UOM Mapping**: Used to define how vendor-specific unit of measure (UOM) codes are mapped to standard UOM codes. For example, a unit of measure (such as kWh) may have different codes (and names) in different head-end systems. UOM mapping allows the measurements of the same types of UOMs from different head-end systems to use a single standard code in Smart Grid Gateway, regardless of the head-end system.

  UOM mapping can also map TOU and SQI values for head-end systems that include that information in their unit of measure codes. For example, the Itron OpenWay head-end system may send UOM codes of “kWh ON” or “kWh GEN.” Both of these would be mapped to the standard UOM code of ““Kilowatt Hours.” For the first UOM code, the TOU code could also be mapped to “On Peak,” and for the second the SQI code could be mapped to “Generated.”

  UOM mapping extendable lookup business objects are typically child business objects of the base (D1-owned) device event mapping business object (D1-HeadendUOMLookup).

- **Interval Status Mapping**: Used to define how vendor-specific interval data codes are mapped to standard status codes. For example, a status code (such as "missing" or "outage") may have different codes (and names) in different head-end systems. Status mapping allows the status codes of the same type from different head-end systems to use a single standard code in Smart Grid Gateway, regardless of the head-end system.

Interval status mapping extendable lookup business objects are typically child business objects of the base (D1-owned) device event mapping business object (D1-IntStsCodeToCondMapLookup).

- **Other**: Other types of extendable lookup business objects are used by adapters to define data used by each adapter. For example, if inbound communications sent to a head-end system include an "action code" that impacts how that communication is processed, available "action codes" can be defined in an extendable lookup.

Example: Landis+Gyr (D3)

The Landis+Gyr adapter includes the following extendable lookup business objects

Business Object	Description
D3-DeviceEventMappingLookup	Landis+Gyr Device Event Mapping
D3-HeadendUOMLookup	andis+Gyr UOM Code to Standard UOM Mapping
D3-IntStsCodeToCondMapLookup	Landis+Gyr Interval Status Code to Condition Mapping
D3-LoadActionCodeLookup	Extendable Lookup - Load Action Code

Oracle Business Process Execution Language Composites

Oracle Utilities Smart Grid Gateway adapters use Oracle Business Process Execution Language (BPEL) composites to provide communications between Smart Grid Gateway and head-end and other external systems.This section outlines the types of BPEL composites provided with Smart Grid Gateway adapters, including:

- **Understanding the Communication Process**
- **Adapter BPEL Composites**

> **Note**: This document describes the BPEL composites in terms of how they are used by Smart Grid Gateway adapters. Refer to Oracle Business Process Execution Language documentation for more information about BPEL composites.

Understanding the Communication Process

This section provides a high-level description of the communication process used when executing smart meter commands from a Smart Grid Gateway adapter. See **Understanding the Command Communication Process** on page 7-10 for more deetails concerning how outbound communications, outbound messages, and inbound communications are created.

1. When a smart meter command is executed for a meter, an outbound message is created and sent to the head-end system.

   Note: The specific type of outbound communication and outbound message created are based on the configuration of the processing methods defined for the head-end system service provider.

2. The message is sent (via XAI sender) to a JMS queue monitored by a BPEL process.
3. The BPEL process picks up the message, and transforms it from the Smart Grid Gatewat format into the format used by the head-end system, and deposits the transfrormed message into a JMS queue monitored by the head-end system.
4. The head-end system attempts to execute the command, and sends a response (based on the success or failure of the command). The response message is sent to a JMS queue monitored by a BPEL process.
5. The BPEL process picks up the message, and transforms it from the format used by the head-end system into the format used by Smart Grid Gateway, and deposits the transfrormed message into a JMS queue associated with an XAI Inbound Service.
6. The XAI inbound service creates an inbound communication (which in turn creates a completion event, and updates the associated outbound communication).

Adapter BPEL Composites

Each Smart Grid Gateway adapter includes a set of BPEL composites used for sending and receiving smart meter commands to and from the head-end system. In addition, BPEL composites are used in bulk command processing.

Adapters typically include composites for the following types of commands:

- Device Status Check (meter ping)
- Commission / Decommission
- Remote Connect / Disconnect
- On-Demand Read

BPEL composites perform transformations and other processing on messages sent between the Smart Grid Gateway and the head-end system.

- When sending messages from Smart Grid Gateway to the head-end system, BPEL converts the message from the Smart Grid Gateway format into the format used by the head-end system.
- When sending message from the head-end system to Smart Grid Gateway, BPEL converts the message from the format used by the head-end system into the format used by Smart Grid Gateway.
- Other processing includes passing data related to a command between Smart Grid Gateway the head-end system, and updating Smart Grid Gateway (or the head-end system) as appropriate. For example, commands may have a job ID in the head-end system. In such cases, the job ID might be sent back to Smat Grid Gateway (and stored on the outbound communication) after the command has been issued, but before it has been executed. Refer to the adapter-specific configuration guides for more information about the specific processing provided by each adapter's BPEL composites.

BPEL composites for vendor-specific adapters should NOT be modified or changed. Composites provided with the Adapter Development Kit can be modified to meet the specific requirements of an implementation. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Configuration Guide* for more information.

Example: Sensus (D6)

The Sensus adapter includes the following BPEL composites:

Command	BPEL Composite
Common	Common
On Demand Read	OnDemandRead
Commission / Decommission	CommissionDecommission
Remove Connect / Disconnect	ConnectDisconnect
Meter Ping	DeviceStatusCheck

SGG Adapter Configuration Portal

The SGG Adapter Configuration portal can be used to view configuration information and to access the required components for a head-end service provider. This section provides a high-level overview of the portal, including:

- **Understanding the SGG Adapter Configuration Portal**
- **Setting Up the SGG Adapter Configuration Portal**

Understanding the SGG Adapter Configuration Portal

Each adapter has many components that must be configured in order for the adapter to process usage events and smart meter commands. The SGG Adapter Configuration portal provides a complete picture of what is required for each adapter, and shows which components of the adapter have been set up and which components need to be set up. The SGG Adapter Configuration portal is displayed as a tab on the service provider page for the head-end system service provider for the adapter.

The SGG Adapter Configuration portal is comprised of the following:

- **SGG Configuration Sheet extendable lookup.** The required configurations for a particular adapter are defined on the SGG Configuration Sheet extendable lookup (D1-SGGAdapterConfigSheet). The lookup specifies the components required for the adapter to process usage, events, and commands. For example, the adapter configuration may include extendable lookup business objects, processing roles, inbound/outbound communication business objects, XAI configurations, and so on.
- **SGG Configuration Tracker zone:** The SGG Configuration Tracker zone on the SGG Adapter Configuration portal displays the following:
  - The components defined on the SGG Configuration Sheet extendable lookup.
  - Additional objects related to the components. For example, for processing roles defined on the configuration sheet extendable lookup, the zone displays the processing role along with the outbound business object, outbound message type, and XAI sender for the processing role.
  - Status messages describing the configuration status of components.

  You can click the displayed data and status messages in this zone to navigate to various portals to configure, view, and maintain the components.

Setting Up the SGG Adapter Configuration Portal

Use the following procedure to set up the SGG Adapter Configuration portal for a head-end system service provider:

1. Select **Admin, E, Extendable Lookup.**
2. Enter "%Configuration" in the Description field.
3. Click **Refresh**.
4. Click the “SGG Adapter Configuration Sheet” link.
5. To configure an existing configuration sheet, click the **broadcast** icon.
6. To create a new configuration sheet, click **Add** in the Extendable Lookup Value list zone title bar.
7. Enter a name for the Adapter Configuration. For example, ITRON_CONFIG.

   Note: This value will be referenced on the appropriate head-end system service provider record.

8. Enter a description for the configuration. For example, Itron Adapter Configuration.
9. Enter an Override Description, Detailed Description, and/or Head-End Software Version number, if applicable.
10. Define General Configuration options, including:
    - Common Configuration
    - IMD Processing Configuration
    - Scheduled Read Configuration
    - Device Event Processing Configuration

    Click the help icons for additional information about each configuration option.
11. Define Command Configuration options for each command supported by the head-end system. Click the help icons for additional information about each configuration option.
12. Click **Save**.
13. Add this configuration sheet to the appropriate head-end system service provider.
    a. Select **Admin, S, Service Provider.**
    b. Click the **broadcast** icon to select the head-end system service provider for the configuration sheet you created in the steps above.
    c. Click the **Edit** button in the Service Provider zone to edit the service provider.
    d. Enter the name of the configuration sheet you created in the steps above.

       Note: This is the value defined for the Adapter Configuration field, NOT the description.
    e. Click **Save**.

    The components defined on the SGG Adapter Configuration Sheet extendable lookup are displayed on the SGG Adapter Configuration tab for the service provider.

Appendix A

Glossary

This glossary provides definitions of commonly used terms.

Activity Type

Defines properties common to a specific type of activity.

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 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.

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.

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 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, SPI, number of intervals, value, etc.)

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.

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.

Identify Spikes

Measurement service used to identify spikes in a specified set of interval data based on a spike percentage tolerance.

Identifiers

Names, numbers, or other values used to identify an entity within the system, including devices, measuring components, service points, etc.

Inbound Communication

Communication sent to Oracle Utilities Smart Grid Gateway from an external system, such as a head-end system or edge application

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.

Insert Intervals

Measurement service used to insert one or more intervals into a set of 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 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.

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.

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

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.

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.

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

Measurement service used to set the condition (status) code of a specified set of interval data.

Shift Intervals

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.

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.

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.

Upload Statistics Activities

Activities used to track processing of initial measurement data and device event upload processing.

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.

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 select from a list of VEE groups (referred to as a matrix) whose associated rules are to be executed 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 Rules Referencing Group

A zone that displays a list of VEE rules that reference the VEE group being viewed in the VEE group portal.

Appendix B

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 Reques**t 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 CI_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 C

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 can make 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 C-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:

- I**LM 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:

- F**inal 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: ILM Date = Current Date - (Default Retention Period (1825) - Sub Retention Period (1095)).

Retention Periods and ILM Dates

By default, the ILM date is calculated as follows: ILM Date = Current Date - 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	Superceded 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) Superceded 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 algorithmns' 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 D

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 D-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 D-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
- **OUAF 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 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	• With Time Zone • Without Time Zone (Default behavior)
Event Data Import Date/Time Format	• With Time Zone • Without Time Zone (Default behavior)
Command Response Date/Time Format	• 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	Used when the head-end system sends date/times in the full XSD format (with the UTC offset). Example: 2015-05-04T11:35:55Z+5 The logic assumes the date/times are in installation time zone standard.

Value	Behavior
Without Time Zone (Default behavior)	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. Example: 2015-05-04T11:35:55 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)

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-SHEVTDTTM) 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-SHEVTDTTM) 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	• 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-VALTMTYTZ)

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-VALDOTTZN)

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 intial measurement and device event upload.

Initial Measurement Data Upload - Date/Time Format Conversion

The Lanids+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 intial 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 intial measurement upload.

Initial Measurement Data Upload - Date Time Format Conversion

The MV90 date/time format (MMDDYYhhmmss) ia 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 intial 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 with 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.