Oracle Utilities Meter Data Management

Configuration Guide

Release 2.0.1 Service Pack 8

E18293-05

October 2012

ORACLE®

Oracle Utilities Meter Data Management/Meter Data Management Installation and Configuration Guide, Volume 1, Release 2.0.1 Service Pack 8

E18293-05

Primary Author: Lou Prosperi

Contents

Chapter 9

Chapter 10

Chapter 11

Chapter 1

Overview

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

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

What Is This Book?

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

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

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

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

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

This guide includes the following chapters:

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

- **Chapter 10**: **Usage Groups and Usage Rules** provides an overview of the usage calculation process, along with technical details concerning related maintenance and business objects.
- **Chapter 11**: **TOU Maps and Dynamic Options** provides an overview of TOU maps and dynamic options and how they are used in the system, along with technical details concerning related maintenance and business objects.
- **Chapter 12**: **Usage Transactions** provides additional information about the usage calculation process, along with technical details concerning usage transaction-related maintenance and business objects.
- **Chapter 14**: **Batch Processing** provides a list of the base package batch controls provided with the system.
- **Chapter 15**: **Integrating Oracle Utilities Meter Data Management with Other Systems** provides a description of how Oracle Utilities Meter Data Management can be integrated with other systems.
- **Chapter 16**: **Sample Implementation** provides a high-level description of the steps involved in configuring Oracle Utilities Meter Data Management in a simple example implementation.
- **Appendix A**: **Measurement Services** provides a list of base package measurement services use by VEE rules and functions.
- **Appendix B**: **Glossary** is a list of commonly used terms.
- **Appendix C**: **Base Package Configuration Objects** provides lists of base package configuration objects that can be leveraged during an implementation.

Other Documentation

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

Installation Documentation

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

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

User Documentation

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

- Oracle Utilities Application Framework Business Process Guide
- Oracle Utilities Application Framework Administraton Guide
- Oracle Utilities Meter Data Framework User's Guide
- Oracle Utilities Meter Data Management User's Guide

Supplemental Documentation

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

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

Embedded Help

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

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

Online Help

Oracle Utilities Meter Data Management also 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 Meter Data Framework: Describes the features and functions provided in the meter data framework (D1)
- Oracle Utilities Meter Data Management: Describes the features and functions provided in the meter data management application (D2)

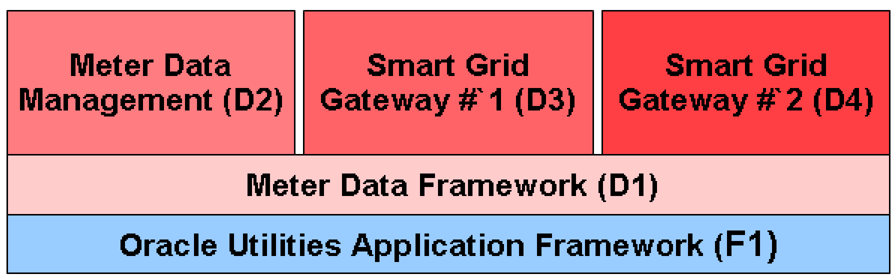
Architecture Overview

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

Oracle Utilities Meter Data Management comprises the following functional areas:

- Device Management: the maintenance of physical meters in the field
- Device Installation: the maintenance of service points and the installation of meters in the field. This includes the means of registering outside systems to Oracle Utilities Meter Data Management for provider/consumer-specific processing of meter events and activities
- Device Communication: the maintenance of communications between Oracle Utilities Meter Data Management and head-end systems, including import of usage and events, 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
- Usage Management: the engine that calculates billable usage recorded on devices, applying factors and dividing the usage into configurable time of use periods

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



Oracle Utilities Application Framework Configuration Tools

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

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

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

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

“Lite” Business Objects

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

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

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

Data Areas

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

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

See **Appendix C**: **Base Package Configuration Objects** for a list of the base package data areas provided with Oracle Utilities meter data framework and Oracle Utilities Meter Data Management.

Algorithms

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

Custom algorithms allow implementers to modify how the system responds to certain system events. The system provides system where the custom algorithms can be invoked instead of the base package algorithms provided with the system. For instructions on creating custom algorithms and algorithm types, see the 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 Meter Data Management system uses naming conventions to identify and distinguish entities that belong to different Oracle applications. These conventions can help you locate entities and understand their context.

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

- All Oracle Utilities Application Framework entities start with "F1"
- All Oracle Utilities meter data framework entities start with "D1"
- All Oracle Utilities Meter Data Management entities start with "D2"

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

Configuration Process Overview

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

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

Basic Configuration Steps - Design Your Business Objects

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

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

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

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

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

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

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

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

Basic Configuration Steps - Create Your Business Objects

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

1. Create configuration objects

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

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

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

2. Create the business object

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

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

Basic Configuration Steps - Create Portals and Zones

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

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

Basic Configuration Steps - Create Master Data

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

1. Create admin “type” data.

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

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

2. Create instances of the data.

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

Admin “Type” Objects.

Object	Admin “Type” Object
Activity	Activity Type
Communication	Communication Type
Contact	Contact Type
Device	Device Type
Device Configuration	Device Configuration Type
Device Event	Device Event Type
Dynamic Option	Dynamic Option Type
Measuring Component	Measuring Component Type
Service Point	Service Point Type
TOU Map	TOU Map Type
Usage Subscription	Usage Subscription Type

Chapter 2

General Configuration

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

- **Installation Options**
- **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 Configuration is a source 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 3-7 for more information on when to set up this record.

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

Meter Data Framework Master Configurations

The following master configurations are used by products that leverage the Oracle Utilities Meter Data Framework, including Oracle Utilities Meter Data Management.

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.

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.

MDM-Specific Business Intelligence Master Configuration

The MDM-Specific Business Intelligence Master Configuration is primarily used to configure the information that will be used in the extraction of data for business intelligence.

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.

Self Service Master Configuration

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

- **Processing Scripts**: Used to retrieve information about measuring components, service points/device configurations, and usage subscriptions
- **Service Tasks**: Used to define self-service tasks available to Oracle Utilities Self Service users.
- **Supported Scalar Usage Groups**: Used to define the scalar usage groups supported when retrieving usage from Oracle Utilities Meter Data Management for access by Oracle Utilities Self Service users.
- **Supported Interval Usage Groups**: Used to define the scalar usage groups supported when retrieving usage from Oracle Utilities Meter Data Management for access by Oracle Utilities Self Service users.

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

Meter Data Management Master Configurations

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

MDM Master Configuration

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

Timeliness Master Configuration

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

Chapter 3

Setting Up Admin Data

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

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

Understanding Admin Data

This section describes the admin data used by the Oracle Utilities meter data framework and Oracle Utilities Meter Data Management.

General Admin Data

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

Exception Types

Exception types define the properties common to many exceptions.

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

Factors

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

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

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

Service Providers

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

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

Service Quantity Identifiers

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

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

Service Types

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

Time of Use

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

Units of Measure

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

Attributes used to define units of measure include the following:

- **Service Type**: The type of service (electric, gas, etc.) measured by the UOM
- **Decimal Positions**: The number of decimal places used when presenting a quantity for this UOM in Usage service quantities
- **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 this type, including the valid types of measuring components that can be configured for device using configurations of this type.

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.

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.

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).

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.

VEE Rule Admin Data

VEE rule admin data include VEE groups and VEE rules.

VEE Groups

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

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

VEE Rules

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

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

Attributes used to define VEE rules typically include the following:

- **Basic Information**: Basic information about the VEE rule, including its name and description, the VEE group to which the rule belongs, the sequence of the rule within the group, the category, and start and end dates. This information is standard for most VEE rules.
- **Parameters**: The parameters used by the rule. Parameters are specific to each rule.
- **Exception Types and Severity**: Details about how to handle exceptions, including the Exception Type and Exception Severity for exceptions created by the rule.

VEE Rule Eligibility Criteria

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

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

Usage Management Admin Data

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

Dynamic Option Types

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

TOU Groups

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

TOU Map Templates

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

Attributes used to define TOU map templates include the following:

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

TOU Map Types

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

Attributes used to define TOU map types include the following:

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

Usage Subscription Types

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

Attributes used to define usage subscription types include the following:

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

Usage Groups

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

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

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

Usage Rules

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

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

Attributes used to define usage rules typically include the following:

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

Usage Rule Eligibility Criteria

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

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

Admin Setup Reference Tables

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

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

Setup Sequence

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

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

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

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

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

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

Administration Setup and Maintenance

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

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

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

Application Framework Setup

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

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

Oracle Utilities Meter Data Management Setup

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

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

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

Chapter 4

Devices, Measuring Components, and, Device Configurations

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

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

Understanding Devices, Measuring Components, and Device Configurations

This section provides an overview of devices, measuring components, and device configurations, including how they are used in the meter data framework 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.

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 meter data framework 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 term Seconds-per-interval (SPI) is 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

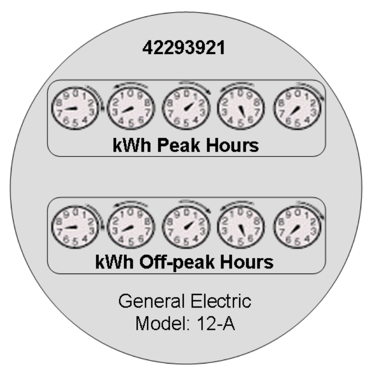
Measuring Component Measurements

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

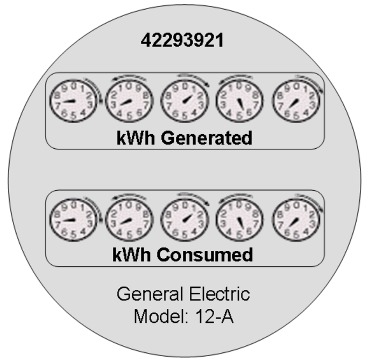
- **Unit of Measure**: The unit of measure for the quantity being recorded. Examples include kilo-watt hours (kWh), kilo-watts (kW), therms, cubic feet (CCF), temperature (Farenheit or Celsius), etc.
- **Time of Use**: Modifiers for a given unit of measure that indicate a period of time during which 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 SPI (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 SPI of 900 (15 minutes) 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
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 SPI), 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.

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.

Meter Data Framework Base Package Device Business Objects

The meter data framework 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-ManualMeter	Manual Meter Instances of this business object represent individual manual meters defined in the system.
D1-SmartMeter	Smart Meter Instances of this business object represent individual smart meters defined in the system.

The meter data framework 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

The meter data framework 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:

- **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 Decomission.

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

Option/Field	Description
Maintenance Object	D1-DEVICE (Device)
Application Service	D1-SMARTMTRBOAS (Smart Meter BO)
Instance Control	Allow New Instances

Options	• Install Event BO: D1-SmartMeterInstallEvent (Smart Meter Installation Event)
	• Synchronization Add BO: D1-SynchronizationAddBO (Device Synchroniation 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.

Meter Data Framework Base Package Device Configuration Business Objects

The meter data framework 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 meter data framework 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 meter data framework 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.

Meter Data Framework Base Package Measuring Component Type Business Objects

The meter data framework base package includes the following measuring component type business objects:

Business Object Name	Description
D1-ActivityStatsAggType	Activity Statistics Aggregator Type Instances of this business object represent individual activty statistics aggregator measuring component types defined in the system. Activty 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 Regiters 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 meter data framework base package includes the following “lite” measuring component type business objects:

Business Object Name	Description
D1-MCTypeEstimateParmsLiteBO	MC Type Periodic Estimation LITE
D1-MCTypeLite	MC Type Lite

Business Object Name	Description
D1-MCTypeMainLite	Measuring Component Type LITE
D1-MCTypeParentLITE	Measuring Component Type Parent LITE
D1-MeasuringCompTypeLite	Measuring Component Type LITE

The meter data framework 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)
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)

Option	Description
Algorithms	• Calculate Interval Consumption: D1-IN-CNSUMP (Calculate Interval Consumption) • Validation: D1-INTSCRVAL (Validate List of Scratchpad Measuring Component Types) • Validation: D1-DEFTOUVAL (Validate Interval Size of TOU Map for Display) • Validation: D1-ELPRFTVAL (Validate Factors for Eligible Profile Factors) • Validation: D1-FNVALOVAL (Validate Final Values Overlay Profiles) • Validation: D1-EVTBARVAL (Validate Event Bar Profiles) • Validation: D1-HRDATAVAL (Hours of Data to Display Validation)

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

Meter Data Management Base Package Measuring Component Type Business Objects

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

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

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

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

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

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

Measuring Components In Detail

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

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

Maintenance Object - D1-MEASRCOMP

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

Option/Field	Description
Maintenance Object	D1-MEASRCOMP
Description	Measuring Component
Service Name	D1-MEASRCOMP (Measuring Component Maintenance)
Tables	• 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.

Meter Data Framework Base Package Measuring Component Business Objects

The meter data framework 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 saggregator measuring components defined in the system.
D1-PayloadStatsAggIMD	Payload Statistics Aggregator - IMD Instance of this business object represent individual measurement payload statistics saggregator 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 meter data framework base package includes the following “lite” measuring component business objects:

Business Object Name	Description
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 meter data framework 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.

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Business Object</td><td>D1-IntervalChannel</td></tr>
<tr><td>Description</td><td>Interval Channel</td></tr>
<tr><td>Maintenance Object</td><td>D1-MEASRCOMP (Measuring Component)</td></tr>
<tr><td>Application Service</td><td>D1-MEASURINGCOMP (Measuring Component MO)</td></tr>
<tr><td>Instance Control</td><td>Allow New Instances</td></tr>
<tr><td>Options</td><td>
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)

Display UI Map: D1-IntervalChannelDisplay (Interval Channel - Display)
Portal Navigation Option: d1mcTabMenu (Measuring Component)
Display Map Service Script: D1-RtMcIcChs (Interval Channel - Retreive Details for Display)
Maintenance UI Map: D1-IntervalChannelMaint (Interval Channel - Maintenance)
</td></tr>
</table>

Option	Description
Algorithms	• Periodic Estimation: D1-CRIMTODO (Create Initial Measurement and To Do Entry) • Information: D1-SMTMCINFO (Measuring Component for Smart Devices - Information) • Pre-Processing: D1-DIMDCPRE (Delete Initial Measurement Data of Measuring Component. • Validation: D1-RELMCCVAL (Validate Consumption Reference Measuring Component) • Validation: D1-INTMCVAL (Check if Measuring Component's Type is Interval)

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

Meter Data Management Base Package Measuring Component Business Objects

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

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

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

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

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

Base Package Interval Initial Measurement Functions

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

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

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

Base Package Measuring Component Consumption Functions

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

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

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

Configuring Devices, Measuring Components, and Device Configurations

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

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

Configuring Custom Devices

Configuring custom devices involves the following steps:

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

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

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

Configuring Custom Device Configurations

Configuring custom device configurations involves the following steps:

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

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

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

Configuring Custom Measuring Component Types

Configuring custom measuring component types involves the following steps:

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

2. Create the custom measuring component type-related configuration objects required for your business objects, including:

   System Events: Create algorithms for the following system events:

   - Calculate Interval Consumption (for interval and scalar measuring component types)
   - Calculate Scalar Consumption (for scalar measuring component types)

   Options: Create data as appropriate for the following options used when creating measuring component types:

   - Value Identifiers: Value Derivation Algorithms (based on the D1-DERIVAQTY Algorithm Type)
   - VEE Groups
   - Profile Factors
   - Scratchpad Measuring Component Types
   - Display Properties:
     - Event Bar Profiles: Values for the D1-360EventBarProfile extendable lookup
     - Final Values Overlay Profiles: Values for the D1-FinalValuesOverlayProfile extendable lookup.

3. Create your measuring component type business objects, referencing the configuration objects created above as appropriate.
4. Set up admin records that define the measuring component types you will use in your implementation.

Configuring Custom Measuring Components

Configuring custom measuring components involves the following steps:

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

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

   - Estimation Initial Measurement Data BO
   - Interval Initial Measurement Function
   - Manual Override IMD BO
   - Measuring Component Consumption Function
   - Synchronization Add BO

3. Create your measuring component business objects, referencing the configuration objects created above as appropriate.

Chapter 5

Service Points and Device Installation

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

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

Understanding Service Points and Device Installation

This section provides an overview of entities related to the installation of devices (service points, contacts, install events, service providers, and others) how they are used in the meter data framework 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

An Aside: No Premise Object Exists

Oracle Utilities Meter Data Management (and other related meter data products) is not 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.

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.

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 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.

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 a processing method that specifies the type of initial measurement data to create for devices (and their related measuring components) based on measuring component type.

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 measuremeny 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.

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 descrived 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_CHAR (Service Point Characteristics) - Child • D1_SP_CONTACT (Service Point Contact) - 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.

Meter Data Framework Base Package Service Point Business Objects

The meter data framework 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 meter data framework base package includes the following “lite” service point business objects:

Business Object Name	Description
D1-SPLite	Service Point Lite
D1-ServicePointODMBORead	Service Point ODM BO to Read
D1-ServicePointParentLITE	Service Point Parent LITE

The meter data framework 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:

- **Process Measurement Cycle**: This system event specifies the algorithm invoked by the Process Measurement Cycle Schedule batch process. This batch process looks for measurement cycle routes that are scheduled to be processed and populates them on the measurement cycle schedule. Once the schedule is prepared, the batch process invokes this algorithm for each device in every route that is ready for processing. The Algorithm Entity for these algorithms is “Service Point (BO) - Process Measurement Cycle.” The base package provides the following algorithm types/algorithms for use with this system event:

Algorithm Type / Algorithm	Description
D1-CRMRDACT	Create Meter Read Download Activity (Creates a meter read download activity for a service point. The service point must have a measurement cycle and route whose schedule indicates a schedule type of “Meter Read Download”)
D2-CRUSGTRN	Create Usage Transaction (Creates a usage transaction for every active usage subscription linked to the service point. The business object used to create the usage transaction is determined from the processing method of the usage subscription's service provider.)

Use the Algorithm Type / Algorithm portal and the Application Viewer to view more details about these algorithms and algorithm types.

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) • Display UI Map: D1-SPDisplay (Service Point - Display) • 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 - Retreive Details for Display) • Maintenance UI Map: D1-SPMaint (Service Point - Maintenance)
Algorithms	• Process Measurement Cycle: D1-CRMRDACT (Create Meter Read Download Activity) • 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)
Lifecycle	• Active (Initial) • Inactive (Final)

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

Meter Data Management Based Package Service Point Business Objects

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

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

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

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

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.

Meter Data Framework Base Package Contact Business Objects

The meter data framework 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 meter data framework base package includes the following “lite” contact business objects:

Business Object Name	Description
D1-ContactLITE	Usage Transaction To Do Contact LITE

Business Object Name	Description
D1-ContactODMBORead	Contact ODM BO to Read

The meter data framework 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) • Display UI Map: D1-BusinessDisplay (Business - Display) • Portal Navigation Option: d1contctTabMenu (Contact) • Display Map Service Script: D1-RtBusDtl (Contact - Retreive Details for Display) • Maintenance UI Map: D1-BusinessMaint (Business - Maintenance)
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.

Meter Data Framework Base Package Install Event Business Objects

The meter data framework 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-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 meter data framework 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 meter data framework 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
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 - Retreive Details for Display) • Maintenance UI Map: D1-SmartMeterInstallEventMaint (Smart Meter Install Event - Maintenance)

Option	Description
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 Device the 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 Info)
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.

Meter Data Framework Base Package Service Provider Business Objects

The meter data framework 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.
D1-ServiceProvider	Service Provider (generic service provider business object used as a parent business object for all other service provider business objects)

The meter data framework base package includes the following “lite” service provider business objects:

Business Object Name	Description
D1-MarketParticipantLite	Market Participant Lite

The meter data framework 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

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
Application Service	D1-SVCPROVIDER (Service Provider MO)
Instance Control	Allow New Instances

Option	Description
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_METH (Processing Method) - Primary • D1_PROC_METH_CHAR (Processing Method Characteristics) - Child • D1_PROC_METH_L (Processing Method Language) - Child

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

Meter Data Framework Base Package Business Objects

The meter data framework 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-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-HowToSendActInformation	How to Send Activity Related Informatoin (used to define how activity-related information is sent 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)

The meter data framework 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 service provider business objects.

Business Object Options

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

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

Lookup FIeld Value	Description
D1AM	Obtain AMI Device Identifier
D1BR	Bulk Response
D1DA	Activity Notification
D1DC	Device Commission
D1DD	Device Decommission
D1DM	Device Event Mapping
D1DS	Device Status Check

Lookup Field Value	Description
D1EP	Event Processing Default Configuration
D1FR	Response - Fail
D1IM	Initial Measurement Creation
D1IN	On-Demand Read (Interval)
D1LC	Load Check
D1MS	Multi-Device Status Check
D1RC	Remote Connect
D1RD	Remote Disconnect
D1RM	Retrieve Meter
D1RR	Response - Received
D1SC	On-Demand Read (Scalar)
D1SD	Send Device Event
D1SQ	SQI Translation
D1SR	Response - Success
D1TU	TOU Translation
D1TZ	Time Zone Translation
D1UM	UOM Translation
D1VC	Verify Commission

System Events

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

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

Algorithm Type / Algorithm	Description
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)

Algorithm Type / Algorithm	Description
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-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)
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-OMTDCT / D1-OMTDCT	Outbound Message Type Differs By Communication 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))

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 Measuring Component 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

Option	Description
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.

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

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

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

Business Object Options

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

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

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

System Events

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

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

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

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

Configuring Service Point and Device Installation Objects

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

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

Configuring Custom Service Points

Configuring custom service points involves the following steps:

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

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

   - Process Measurement Cycle
3. Create your service point business objects, referencing the configuration objects created above as appropriate.
4. Set up admin records that define the service point types you will use in your implementation.

Configuring Custom Contacts

Configuring custom contacts involves the following steps:

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

Configuring Custom Install Events

Configuring custom install events involves the following steps:

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

Configuring Custom Service Providers

Configuring custom service providers involves the following steps:

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

   Processing Methods: Create processing method business objects for use with each service provider.
3. Create your service provider business objects, referencing the configuration objects created above as appropriate.

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

Configuring Custom Processing Methods

Configuring custom processing methods involves the following steps:

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

   System Events: Create algorithms for the following system events:

   - Determine Processing Method(s)
3. Create your processing method provider business objects, referencing the configuration objects created above as appropriate.

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

Chapter 6

Measurement Data

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

- **Understanding Initial Measurement Data and Final Measurements**
- **Initial Measurement Data In Detail**
- **Measurements In Detail**
- **Configuring Initial Measurements and Measurements**

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 meter data framework and Oracle Utilities Meter Data Managemen, 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).

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 incokming 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 approrpriate 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 consitute an undercount.

- **Overcount**: An overcount occurs when an interval initial measurement contains more interval values than approrpriate 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 consitute 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

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.

Validation, Editing, and Estimation

Once received into the system, initial measurements are subject to validation, editing, and estimation. This process involves the following:

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

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 mutipliers 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 the 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 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. The measuring component has a multiplier other than 1.

   In this case, the Post VEE value is equal to the Pre VEE value times the multiplier.
2. The installation event has a constant other than 1.

   In this case, the Post VEE value is equal to the Pre VEE value times the installation constant.
3. 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
4. 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
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.

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).

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 creates a initial measurement via estimation. 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 “Period 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 occurs at different points in time for interval and scalar measuring components:

- For interval measuring components, a dedicated batch process exists to initiate creation of estimated initial measurements
- For manually read scalar measuring components, a usage calculation rule creates the initial measurement if it cannot find a measurement and it has been given permission to estimate. (Note: the information that follows applies primarily to interval measuring components).

The batch process that detects missing consumption for interval measuring components uses two elements on interval measurinng component type:

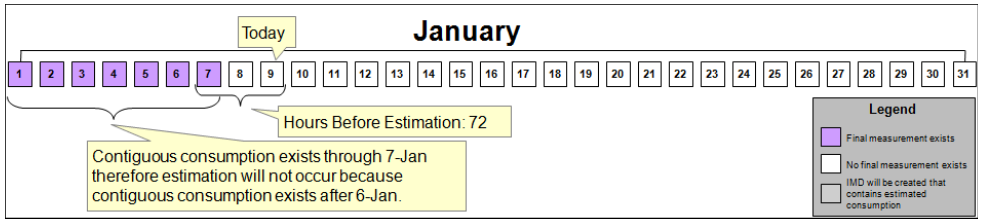
- **Hours Before Estimation**: the number of hours after the End Date and Time of the most recent measurement that must pass before the measuring component is considered due for estimation. For example, if set to 72 hours, estimation will only take place 72 hours after the End Date and Time of the latest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

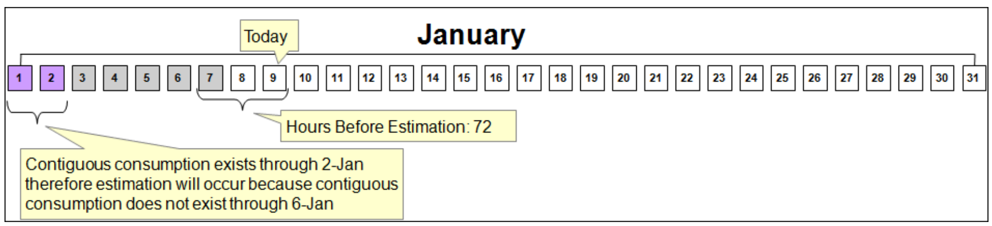
If the measuring component does NOT have contiguous final measurements on/after a date and time equal to the current date/time minus the “Hours Before Estimate” value, the measuring component is estimated. If the measuring component is subject to estimation, measurements will be estimated through the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

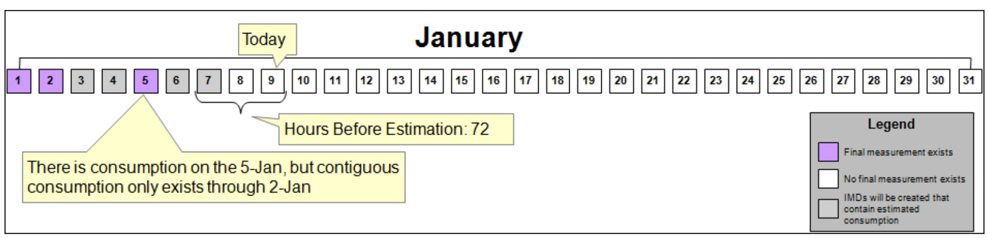
Example 1: On January 9, if there is contiguous consumption through January 7, estimation would not occur because consumption exists after January 6 (3 days prior to January 9). The diagram below illustrates this example:



Example 2: On Janury 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 Janury 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:



Estimation Algorithms

Estimation for interval 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 (in most cases, interval measuring components) have a “Period 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) device business object. The “Periodic Estimation” monitor algorithm can be triggered by the “Smart Meter State Monitor Process” batch process (D1-SMMTR).

The “Periodic Estimation” System Event algorithm determines 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, this algorithm is the “Create Initial Measurement and To Do Entry” algorithm (D1-CRIMTODO). 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 base package includes this algorithm on the Period Estimation system event state of the Interval Channel (D1-IntervalChannel) measuring component business object.

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 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 perform a sum check to validate the resulting interval measurement against measurements from a consumption reference measuring component. In this example, the sum check 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).

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.

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 meter data framework 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 meter data framework 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 meter data framework 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 meter data framework 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 intial measurement; the meter data framework 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 fanagement 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

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
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
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.

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 inteval 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 inteval 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 inteval 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:

- **IMD Status Extendable Lookup**: This option defines an extendable lookup that can be used to define different statuses which can apply to initial measurement data created from this business object.
- **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
D1GA	IMD Seeder
D1IL	Initial Load
D1MO	Manual
D1ES	Estimation

- **Interval Status Mapping to Condition with Priority**: 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.

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	• Initial Measurement Data Type: D1IL (Initial Load) • 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 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.

Meter Data Management Base Package Initial Measurement Data Business Objects

The Oracle Utilities Meter Data Management base package includes the following initial measurement data business objects:

Business Object Name	Description
D2-IMDHoursLateLite	IMD Hours Late Lite

Measurements In Detail

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

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

Maintenance Object - D1-MSRMT

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

Option/Field	Description
Maintenance Object	D1-MSRMT
Description	Measurement
Service Name	D1-MEASUREMENT (Measurement Maintenance)
Tables	• 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)

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

Meter Data Management Base Package Measurement Business Objects

The Oracle Utilities Meter Data Management base package includes the following aggregated measurement business objects:

Business Object Name	Description
D2-MeasuredQuantityMsrmt	Measured Quantity Measurement Instances of this business object represent individual measured quantity aggregated measurements stored in the system.
D2-QualityCountMsrmt	Quality Count Measurement Instances of this business object represent individual quality count aggregated measurements stored in the system.
D2-TimelinessCountMsrmt	Timeliness Count Measurement Instances of this business object represent individual timeliness count aggregated measurements stored in the system.
D2-TimelinessQualityMsrmt	Timeliness Quality Measurement Instances of this business object represent individual timeliness quality aggregated measurements stored in the system.

Configuring Initial Measurements and Measurements

This section provides high-level overviews of the steps involved in configuring custom objects to define initial measurements and (final) measurements. 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 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.

Chapter 7

Validation, Editing, and Estimation

This chapter provides descriptions of validation, editing, and estimation groups and rules, including:

- **Understanding Validation, Editing, and Estimation**
- **VEE Groups In Detail**
- **VEE Rules In Detail**
- **VEE Eligibility Criteria In Detail**
- **VEE Exceptions In Detail**
- **Configuring VEE Groups, Rules, Eligibility Criteria, and Exceptions**

Understanding Validation, Editing, and Estimation

This section describes the validation, editing, and estimation (or VEE) process used by the meter data framework and related products, including Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway. This section includes:

- **Overview of the Validation, Editing, and Estimation Process**
- **VEE Rules and VEE Groups**
- **VEE Exceptions**

Overview of the Validation, Editing, and Estimation Process

As noted in **Chapter 6**: **Measurement Data**, once received into the system, initial measurements are subject to validation, editing, and estimation. This process involves the following:

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

The values recorded in an initial measurement can change during this process, and exceptions can be raised if initial measurements are wrong in some way (such as out-of-tolerance quantities, incorrect values, etc.).

Beyond raising an exception, the VEE process can also transition an initial measurement to an Exception state if it detects a problem that it is not able or allowed to correct.

VEE Rules and VEE Groups

The specific validation, editing, and estimation processing performed on initial measurement data is defined in individual VEE rules, each performing a specific set of validation logic. Examples of VEE rules include:

- Interval Size Validation: Checks to ensure that the interval size of the incoming initial measurement data matches the value defined in the measuring component type
- Multiplier Check: Checks to ensure that the device multiplier value of the incoming initial measurement data matches the multiplier value stored on the measuring component
- Unit of Measure Check: Checks to ensure that the Unit of Measure (UOM) of the incoming initial measurement data matches the UOM specified on the measuring component's type

The base package contains many VEE rules you can use in your implementation, and you can also create your own custom VEE rules.

Some VEE rules create exceptions if the initial measurement data doesn't fall within parameters specified by the rule. Other rules override measurements, changing measurement values as dictated by the rule's parameters. Some rules can both create exceptions and override the measurement as part of a single process. By convention, VEE rules change the Post VEE quantities of initial measurement data, but VEE rules can change ANYTHING on an initial measurement.

VEE Groups

VEE groups are collections of VEE rules that are applied to initial measurement data. During the VEE process, the system executes the VEE rules defined in each VEE group. The rules within a

VEE group are defined in a specific sequence, allowing control over the order in which the rules are executed.

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

Effective Dates

Every VEE rule has an effective period. Rules will only be applied if the initial measurement's start date is within the rule's effective period. For example, an Interval Spike Check rule with a Start Date of 11/15/2010 will only be applied if the start date of the initial measurement is on or after 11/15/2010.

This allows you to update the specifics of a rule without removing the previous version of the rule. For example, you might change the tolerance of an Interval Spike Check rule from 1.2 to 1.5 as of a certain date. However, for initial measurement data for the period prior to the change, you would want to use the tolerance for the original version of the rule (1.2) instead of the new tolerance (1.5).

Eligibility Criteria

Each VEE rule may optionally have eligibility criteria that controls if the rule is applied. This feature can greatly reduce the number of VEE groups you need to create, because it allows a single VEE group to have conditional VEE rules based on eligibility criteria (rather than requiring a distinct VEE group for every combination of VEE rules).

For example, you might create a rule that compares interval consumption against related scalar consumption (such as might be the case with a device with both interval and scalar measuring components). This rule might use eligibility criteria that specifies that the rule is only applied if the initial measurement's measuring component has a corresponding scalar register measuring component.

Another example might be a rule that compares the current consumption against standard consumption for measuring components of a certain type for the first six months after installation. You might create eligibility criteria that specifies that the rule is only applied if the measuring component's device configuration has been installed at a service point for less than six months.

Generic Utility Rules - Tools For Creating VEE Groups

While most VEE rules are used to validate the usage (or some other attribute) in an initial measurement, Oracle Utilities meter data framework also provides a number of "generic utility" VEE rules that can be used when configuring VEE groups. These include:

- Referred VEE Group Rules
- VEE Group Matrix (Factor) Rules
- Exception Handler Rules
- Successful Termination Rules

Execute VEE Group Rules - Reusing Groups Of Rules

A common situation in many implementations is in which several rules are to be applied to multiple different types of measuring components. For example, you might want to perform device identifier validations, multiplier checks, and UOM checks on all measuring components.

One way to meet this requirement would be to repeat these three rules in multiple VEE groups. However, this solution becomes hard to maintain if changes to the rules are required (or if new "global rules" are introduced) as each group would have to be updated.

Instead of this, you can create a VEE rule that executes the rules in a referenced VEE group. Rules of this type are called Execute VEE Group rules. Rules of this type can have effective dates and eligibility criteria, just like all VEE rules.

Using the example above, you could create a group called “Rules for All MCs” that contains a device identifier validations rule, a multiplier check rule, and a UOM check rule, and then reference the “Rules for All MCs” group in a Execute VEE Group rule.

Execute VEE Group rules can be “nested.” That is, a group executed by a Execute VEE Group rule can, in turn, execute the rules in another group, and so on.

VEE Group Matrix (Factor) Rules - Using Factors To Implement Dynamic VEE Groups

Another situation likely to occur in many implementations is where specific rules may need to be applied to measurement data based on specific criteria, such as geography. For example, some geographic territories may have unique VEE rules in addition to rules that are applied to all geographic territories.

This requirement could be implemented using eligibility criteria, such as only applying a rule (or a group of rules) if the service point for an initial measurement is located in a hot summer area. If there are limited number of these unique rules, this solution is suitable, but if there are many territories and each territory has several unique rules, an implementation of this sort would become hard to maintain (and slow to execute) as the VEE group would have many rules with varying eligibility criteria.

Instead of this, you can create a VEE rule that dynamically executes another group's rules based on specific conditions. For example, the VEE process can be configured to execute different VEE rules based on where the service point is located, or the number of tamper events in the last 6 months, or the type of customer, or the meter's head-end system, etc. Rules of this type are called VEE Group Matrix (Factor) rules. These rule and can have effective dates and eligibility criteria, just like all VEE rules.

Factors are used to implement the dynamic selection of VEE group (note the term factor is intentionally generic as factors can be used for other purposes). Factors used for these rules have a Factor Class of “VEE group,” and use some unique rules:

- VEE group factors reference a characteristic type (with pre-defined values).
- VEE group factors reference an algorithm that retrieves or derives the value of the characteristic type at runtime.
- Factor values for a VEE group factor are effective-dated pairings of a characteristic value and a corresponding VEE group.

At run time, the rule retrieves / derives the characteristic value for the factor's characteristic type and then finds the VEE group associated with the respective characteristic value.

Factors can be related to any real or dynamic attribute, so rules of this type are very flexible. For example:

- **Real Attribute**: you could create a rule that executes a VEE group based on the head-end system of the device.
- **Dynamic Attribute**: you could create a rule that executes a VEE group based on the number of tamper events linked to the measuring component in the last 180 days, executing one group if there are 6-10 events (a characteristic value of 6-10), and another if there are more than 10 events (a characteristic value of 10+). The number of tamper events is dynamically calculated at execution time and is compared to the characteristic values defined for the factor, and executes the appropriate VEE group. In this example, if the count of tamper events was anything less than six, no VEE group would be executed.

Exception Handler Rules

Exception Handler rules are described in the section below on VEE exceptions.

Successful Termination Rules

Successful Termination rules are described in the section below on VEE exceptions.

VEE Roles

Initial measurement data can come from different sources, such as a head-end system or estimation processes, or it can be manually created by a user (to override or estimate consumption). Measurement data from these different sources might use different VEE rules. For example:

- Initial measurements sent a head-end system might use strict VEE rules
- Initial measurements created by a user (to override or estimate consumption) may use less strict rules
- Initial measurements created by the system to estimate consumption have very few (if any) VEE rules

Applying different numbers and types of VEE rules based on the source of the initial measurement data could be implemented using eligibility criteria (e.g., only apply a rule if the initial measurement data's source is X) or factor-based rules (e.g., the factor's characteristic is the initial measurement data's source), but both of these techniques are potentially difficult to maintain if there are many source-dependent rules.

Instead of that approach, you can define different VEE groups for different source, or roles. The three base package roles are:

- **Estimation**: Used for initial measurement data estimated by the system
- **Initial Load**: Used for initial measurement data received from a head-end system or import process
- **Manual Override**: Used for initial measurement data manually created by a user

A measuring component's Measuring Component Type can define "fallback" VEE groups for each of these roles. In addition, an individual measuring component can specific a VEE group for each role. If the measuring component doesn't have a VEE group specified for a role, the "fallback" VEE group defined for the measuring component type is used.

VEE Exceptions

Each VEE rule defines an exception type and severity that specify how exceptions are tracked by the system. When an initial measurement fails a validation, an exception of the type specified for the failed VEE rule is created. A single initial measurement can have multiple exceptions, one (or more) for each rule the measurement fails. This allows users to see all of the problems detected during the VEE process.

Exception Types

Each exception has an exception type. Exception types allow you to distinguish between different exceptions based on the rule that triggered them. Exception types can be created each VEE rule, at a more general level exception types, such as "Insufficient Data" to be used to signify that a measurement didn't have sufficient data for the VEE rule to execute.

Exception Categories

There are three categories, or severities of exceptions:

- **Info**: Used to highlight something interesting, but not sufficient to cause the initial measurement to be put into the Exception state. Exceptions of this category can be used to report on the frequency of interesting, but not fatal issues.
- **Issue**: Used to report a problem that will prevent the initial measurement from being finalized. Multiple "issue exceptions" can be created during VEE processing. If at least one issue exists after all rules have been applied, the initial measurement is transitioned to the Exception state.

- **Terminate**: Used to report a severe issue that will cause the VEE process to stop and the initial measurement to be transitioned immediately to the Exception state. Only one terminate exception can be issued (as the first one causes VEE to stop on an initial measurement).

Exceptions and To Do Entries

In addition to exceptions, VEE processing can also trigger the creation of To Do Entries related to failed validations.

If Issue or terminate exceptions exist for an initial measurement, a To Do Entry is created when the initial measurement is transitioned to the Exception state. The To Do Type and default To Do Role of this To Do Entry are defined on the Enter system event for the Exception state of the business object used to define the initial measurement.

To Do Entries created in this way can be routed to different roles depending on the exception's message category and number (using the To Do Type's Message Overrides tab).

Exception Handler Rules - Aborting When There Are Too Many Issues

There may be times when an implementation's requirements are to terminate processing for any initial measurement that contains a pre-defined number of exceptions. Exception Handler VEE rules can issue a "terminate exception" if they detect too many exceptions. This is useful when individual exceptions are not sufficient to stop VEE processing.

The criteria used by this rule can simply reference a number of exceptions of a given exception type, or can specify more complex AND/OR criteria that must be satisfied before VEE processing is terminated. For example, processing might terminate when 3 exceptions of one type AND 2 exceptions of another type have been issued, or if 2 exceptions of one type OR 2 exceptions of a different type have been issued.

The terminate exception created by Exception Handler rules can be of a specific exception type. In addition, Exception Handler rules can also create a different type of To Do Type and To Do Role than the default.

Exception Handler rules can be placed at any point throughout a VEE group where each rule can reference different exception types.

Successful Termination Rules

There may be times when an implementation's requirements are to successfully terminate processing for any initial measurement that passes a pre-defined set of validations before accumulating a pre-defined number of exceptions. For example, a set of validation rules can be executed early in the overall sequence of rules that proves that the data is good enough to use, such that no further rules need to be executed. In this case, implementations might want to terminate the VEE process to save on execution time rather than execute further rules that won't ultimately affect the data. This is accomplished through Successful Termination rules.

The criteria used by Successful Termination rules can simply reference a number of exceptions of a given exception type, or can specific more complex AND/OR criteria that must be satisfied before VEE processing is terminated. For example, processing might terminate when less than 3 exceptions of one type AND less than 2 exceptions of another type have been issued, or if less than 2 exceptions of one type OR less than 2 exceptions of a different type have been issued.

Successful termination rules can be placed at any point throughout a VEE group where each rule can reference different exception types.

Available Actions for Initial Measurements with Exceptions

Users have a number of options for dealing with initial measurements with exceptions.

- After correcting the cause of the issues that triggered the exceptions, a user can re-VEE the initial measurement.

- A user can discard the initial measurement.
- A user can edit the Post VEE quantities (if necessary) and manually complete the initial measurement. This will cause final measurements to be created using the contents of the Post VEE quantities.

  Note: No VEE processing is performed on manually completed initial measurement data.

Regardless of the action taken by the user, the system will complete any open To Do Entries that created when the initial measurement entered the Exception state.

Exceptions Are Not Deleted

Note that exceptions are not deleted when an initial measurement is adjusted or corrected. After any issues are corrected or the initial measurement is overridden (or manually completed), the exceptions persist (in the Closed state) for reporting purposes.

VEE Groups In Detail

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

- A description of the D1-VEEGROUP maintenance object
- Lists of the base package VEE group business objects, including "lite" business objects
- A sample VEE group business object (D1-VEEGroup)

Maintenance Object - D1-VEEGROUP

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

Option/Field	Description
Maintenance Object	D1-VEEGROUP
Description	VEE Group
Service Name	D1-VEEGROUP (VEE Group Maintenance)
Tables	• D1_VEE_GRP (VEE Group) - Primary • D1_VEE_GRP_CHAR (VEE Group Characteristics - Child • D1_VEE_GRP_L (VEE Group Language) - Child

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

Meter Data Framework Base Package VEE Group Business Objects

The meter data framework base package includes the following VEE group business objects:

Business Object Name	Description
D1-VEEGroup	VEE Group

The meter data framework base package includes the following additional VEE group business objects:

Business Object Name	Description
D1-VEEGroupBundlingAddBO	Bundling Add BO for VEE Group
D1-VEEGroupPhysicalBO	Physical BO for VEE Group

Example VEE Group - D1-VEEGroup

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

Option/Field	Description
Business Object	D1-VEEGroup
Description	VEE Group
Maintenance Object	D1-VEEGROUP (VEE Group)
Application Service	D1-VEEGROUP (VEE Group MO)
Instance Control	Allow New Instances
Options	• Display UI Map: D1-VEEGroupDisplay (VEE Group - Display) • Portal Navigation Option: d1veegrpTabMenu (VEE Group) • Maintenance UI Map: D1-VEEGroupMaint (VEE Group - Maintenance)

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

VEE Rules In Detail

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

- A description of the D1-VEERULE maintenance object
- Lists of the base package VEE rule business objects, including "lite" business objects
- Details concerning VEE rule-specific configuration options
- A sample VEE rule business object (D2-IntervalSpikeCheck)
- A list of base package VEE rules, including the algorithm / algorithm type and a brief description of each

Maintenance Object - D1-VEERULE

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

Option/Field	Description
Maintenance Object	D1-VEERULE
Description	VEE Rule
Service Name	D1-VEERULE (VEE Rule Maintenance)
Tables	• D1_VEE_RULE (VEE Rule) - Primary • D1_VEE_RULE_CHAR (VEE Rule Characteristics) - Child • D1_VEE_RULE_L (VEE Rule Language) - Child

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

Meter Data Framework Base Package VEE Rule Business Objects

The meter data framework base package includes the following VEE rule business objects:

Business Object Name	Description
D1-DuplicateIMDCheck	Duplicate IMD Check Instances of this business object represent individual Duplicate IMD Check rules defined in the system.
D1-GenericVEERule	Generic VEE Rule (generic VEE rule business object used as a parent business object for all other VEE rule business objects)
D1-VEERuleExceptionHandler	VEE Rule Exception Handler Instances of this business object represent individual exception handler VEE rules defined in the system.

Business Object Name	Description
D1-VEERuleGroupFactor	VEE Group Matrix (Factor) Instances of this business object represent individual VEE Group Matrix (Factor) VEE rules defined in the system.
D1-VEERuleReferredVEEGroup	Execute VEE Group Instances of this business object represent individual Execute VEE Group VEE rules defined in the system.
D1-VEERuleSuccessTermination	Successful Termination Instances of this business object represent individual Successful Termination VEE rules defined in the system.

The meter data framework base package includes the following “lite” VEE rule business objects:

Business Object Name	Description
D1-VEERuleGroupFactorLite	VEE Rule Group Factor LITE
D1-VEERuleParentLITE	VEE Rule Parent LITE

The meter data framework base package includes the following additional VEE rule business objects:

Business Object Name	Description
D1-VEERuleBundlingAddBO	Bundling Add BO for VEE Rule
D1-VEERuleExecReSequence	VEE Rule Execution Resequencing (used when resequencing VEE rules in a VEE group)
D1-VEERulePhysicalBO	Physical BO for VEE Rule

The Oracle Utilities Meter Data Management base package includes the following VEE rule business objects:

Business Object Name	Description
D2-EnsureIMDExistsforSibling	Ensure IMD Exists for Sibling MCs
D2-IntervaAdjustmentFrmScalar	Interval Adjustment from Scalar
D2-IntervalAveragingEstimation	Interval Averaging Estimation
D2-IntervalInterpolationEst	Interval Interpolation Est
D2-IntervalProfileEstimation	Interval Profile Estimation
D2-IntervalReplacementRule	Interval Replacement Rule
D2-IntervalSizeValidation	Interval Size Validation
D2-IntervalSpikeCheck	Interval Spike Check
D2-NegativeConsumptionCheck	Negative Consumption Check
D2-RaiseMissingQuantityExcp	Raise Missing Quantity Exception

Business Object Name	Description
D2-RegisterMultiplierCheck	Multiplier Check
D2-ScalarCalcFromInterval	Scalar Calculation From Interval
D2-ScalarEstimation	Scalar Estimation
D2-ScalarProfileEstimation	Scalar Profile Estimation
D2-ScalarReplacementRule	Scalar Replacement Rule
D2-SumCheck	Sum Check
D2-UOMCheck	Unit of Measure Check
D2-VEERuleHighLowCheck	High/Low Check
D2-ZeroConsumptionCheck	Zero Consumption Check

Configuration Options

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

System Events

VEE rule business objects can make use of the following system events:

- **Apply VEE Rule**: This system event defines the algorithm to use when executing the VEE rule.

Other Options

VEE rules use various parameters and properties. These options are specified when creating VEE rules based on a VEE rule business object, and include the following:

Exception Types

Exception types define the properties common to exceptions. When creating VEE rules, you might create an exception type for each rule. You might also create more general exception types, such as "Insufficient Data" to be used to signify that a measurement didn't have sufficient data for the VEE rule to execute.

Generic Utility VEE Rules

Oracle Utilities meter data framework includes three “generic utility” base package VEE rule types that can be used when configuring VEE groups and rules. This section outlines the configuration options you need to configure before you can create rules of these types.

Execute VEE Group: Referred VEE Group rules reference a VEE group. You must create the VEE group to reference before can create rules of this type.

Exception Handler: Exception Handler rules are used to define options and logic to terminate the VEE process when a set of user configured criteria are met. VEE rules of this type can be included in a group to specify how exceptions are handled for that group, and allow for creation of a single "parent" exception for the group. Exception Handler rules use the following options:

- **To Do Type**: An override To Do Type for To Do Entries created as a result of the rule. This To Do Type is used instead of the default (specified in the Enter algorithm on the Exception state of the initial measurement data business object’s lifecycle).

- **To Do Role**: An override To Do Role for To Do Entries created as a result of the rule. This To Do Role is used instead of the default (specified in the Enter algorithm on the Exception state of the initial measurement data business object's lifecycle).
- **Exception Type**: The Exception Type for exceptions created by this rule.

VEE Group Matrix (Factor): VEE Group Factor rules are used to define business logic to allow reference to a factor (of type VEE group) where the values of the factor are a list of VEE groups. This allows creating a VEE rule that can select from a list of VEE groups (referred to as a matrix) whose rules to execute next. VEE Group Matrix (Factor) rules use the following options:

- **Factor**: The factor referenced by the rule. The factor must have a Factor Class of VEE Group (i.e. it must be based on the VEE group factor business object).
- **Characteristic Type**: The characteristic type referenced by the factor. This characteristic type must be one with pre-defined values.
- **Characteristic Type Values**: Specific values for the characteristic type. These are the values retrieved and evaluated to determine the VEE group whose rules should be executed. These must be values that can be retrieved from some object (device, service point, etc.) related to the measuring component whose initial measurement data is being validated by this rule.
- **Characteristic Source Algorithm**: The algorithm used to retrieve the characteristic value (which in turn determines the VEE group whose rules should be executed). The base package includes the following algorithm types that can be used when creating this algorithm:
    - Factor Characteristic Source - Device (D1-FCSDEVICE)
    - Factor Characteristic Source Measuring Component (D1-FCSMC)
    - Factor Characteristic Source - Service Point (D1-FCSSP)
    - Factor Characteristic Source - Usage Subscription (D1-FCSUS)
- **VEE Groups**: The VEE groups associated with each characteristic value.

Example VEE Rule - D2-IntervalSpikeCheck

The table below lists the details of the D2-IntervalSpikeCheck VEE rule business object.

Option/Field	Description
Business Object	D2-IntervalSpikeCheck
Description	Interval Spike Check
Maintenance Object	D1-VEERULE (VEE Rule)
Parent Business Object	D1-GenericVEERule (Generic VEE Rule)
Lifecycle Business Object	Generic VEE Rule
Application Service	D1-VEERULE (VEE Rule MO)
Instance Control	Allow New Instances
Options	• Display UI Map: D2-IntervalSpikeCheckDisplay (Interval Spike Check - Display) • Portal Navigation Option: d1veerleTabMenu (VEE Rule) • Maintenance UI Map: D2-IntervalSpikeCheckMaint (Interval Spike Check - Maintenance)

Option/Field	Description
Algorithms	• Apply VEE Rule: D2-INTSPKCHK (Interval Spike Check) • Validation: D2-INTSPKVLD (Interval Spike Check Minimum Number of Intervals Validation)

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

Meter Data Management Base Package VEE Rule Business Objects

The Oracle Utilities Meter Data Management base package includes the following VEE rule business objects:

Business Object Name	Description
D2-EnsureIMDExistsforSibling	Ensure IMD Exists for Sibling MCs
D2-IntervaAdjustmentFrmScalar	Interval Adjustment from Scalar
D2-IntervalAveragingEstimation	Interval Averaging Estimation
D2-IntervalInterpolationEst	Interval Interpolation Est
D2-IntervalProfileEstimation	Interval Profile Estimation
D2-IntervalReplacementRule	Interval Replacement Rule
D2-IntervalSizeValidation	Interval Size Validation
D2-IntervalSpikeCheck	Interval Spike Check
D2-NegativeConsumptionCheck	Negative Consumption Check
D2-RaiseMissingQuantityExcp	Raise Missing Quantity Exception
D2-RegisterMultiplierCheck	Multiplier Check
D2-ScalarCalcFromInterval	Scalar Calculation From Interval
D2-ScalarEstimation	Scalar Estimation
D2-ScalarProfileEstimation	Scalar Profile Estimation
D2-ScalarReplacementRule	Scalar Replacement Rule
D2-SumCheck	Sum Check
D2-UOMCheck	Unit of Measure Check
D2-VEERuleHighLowCheck	High/Low Check
D2-ZeroConsumptionCheck	Zero Consumption Check

Base Package VEE Rules Summary

The following table lists the back package VEE rules. Each of these VEE rules is provided as a business object and corresponding algorithm/algorithm type.

VEE Rule	Business Object	Algorithm / Algorithm Type	Description
Duplicate IMD Check	D1-DuplicateIDMCheck	D1-DUPIMDCHK	Checks whether the current IMD is a duplicate of one already received for the same measuring component.
Generic VEE Rule	D1-GenericVEERule	N/A	Parent VEE rule for all other VEE rules, contains basic VEE rule schema elements
VEE Rule Exception Handler	D1-VEERuleExceptionHandler	D1-AVEER-EEH	Allows a user to define a set of exception count criteria that, if met, results in the termination of the VEE process.
VEE Group Matrix (Factor)	D1-VEERuleGroupFactor	D1-AVEER-FCT	Uses the factor configured on the rule to determine a VEE group whose rules are to be executed
Execure VEE Group	D1-VEERuleReferredVEEGroup	D1-AVEER-RFG	Executes the rules for the VEE group configured on the rule
Device Multiplier Check	D2-RegisterMultiplierCheck	D2-REGMULCHK	Checks to ensure that the device multiplier value of the incoming initial measurement data matches the multiplier value stored on the measuring component.
Energy Sum Check	D2-SumCheck	D2-SUM-CHK	Checks the difference between the total consumption of incoming initial measurement data against the total consumption (based on final measurements) for one or more related measuring components on the same device over the same time period.
Ensure IMD Exists for Sibling MCs	D2-EnsureIMDExistsforSibling	D2-ENSIMDMC	Validates that initial measurements exist for all other measuring components associated to the same device configuration as the current measuring component, for the same period of time as the current initial measurement
High/Low Check	D2-VEERuleHighLowCheck	D2-HILO-CHK	Compares consumption of incoming initial measurement data against historical data as a means of assuring the reasonableness of the data.
Interval Averaging Estimation	D2-IntervalAveragingEstimation	D2-INTAVGEST	Estimates missing interval initial measurement data based on averaging historical data for the same measuring component
Interval Interpolation Est	D2-IntervalInterpolationEst	D2-INTINTEST	Estimates missing interval initial measurement data using linear interpolation
Interval Profile Estimation	D2-IntervalProfileEstimation	D2-INTPROEST	Estimates missing interval initial measurement data based on a specified profile (a profile factor)

VEE Rule	Business Object	Algorithm / Algorithm Type	Description
Interval Replacement Rule	D2-IntervalReplacementRule	D2-INTREPL	Determines whether or not an incoming initial measurement will replace any existing measurements and if so, whether or not to allow this to happen.
Interval Size Validation	D2-IntervalSizeValidation	D2-INTSIZVAL	Checks to ensure that the interval size of the incoming initial measurement data matches the value defined in the measuring component type.
Interval Spike Check	D2-IntervalSpikeCheck	D2-INTSPKCHK	Examines incoming initial measurement data to identify intervals with conspicuously high usage relative to surrounding intervals.
Negative Consumption	D2-NegativeConsumptionCheck	D2-NCON-CHK	Checks for negative consumption.
Raise Missing Quantity Exception	D2-RaiseMissingQuantityExcp	D2-RAIMISQTY	Raises an exception if a percentage of the measurements within an initial measurement are marked with a condition code that falls within a specified range
Scalar Estimation	D2-ScalarEstimation	D2-SCALAREST	Estimates missing scalar initial measurement data based on historical data for the same measuring component.
Scalar Replacement Rule	D2-ScalarReplacementRule	D2-SCAREPRL	Identifies whether or not a scalar reading will completely replace an existing measurement and if so, whether or not to allow this to happen.
Unit Of Measure Check	D2-UOMCheck	D2-UOMCHK	Checks to ensure that the Unit of Measure (UOM) of the incoming initial measurement data matches the UOM specified on the measuring component's type.
Zero Consumption Check	D2-ZeroConsumptionCheck	D2-ZEROCNCHK	Checks for zero consumption.

Use the Business Object portal and Algorithm portal (or Application Viewer) to view additional details about these VEE rules. The algorithm types listed above use a set of base package measurement services. See **Appendix A**:**Measurement Services** for a list of available base package measurement services.

Base Package VEE Rule Descriptions

This section provides detailed descriptions of each of the base package VEE rules provided with Oracle Utilities Meter Data Management. The descriptions of the VEE rules below contain the following information

- **Rule Name**: The name of the rule
- **Base Package VEE Rule Business Object**: The base package business object used by the rule
- **Apply VEE Rule Algorithm Type / Algorithm**: The base package algorithm type and algorithm specified in the "Apply VEE Rule" system event for the rule
- **Apply VEE Rule Algorithm Parameters**: The "soft" parameters (if any) used by the algorithm used by the rule.

  Most parameters used by "Apply VEE Rule" algorithms are used to define ranges of measurement conditions used in validation processing (measurement conditions are assigned a numerical value as well as a description). The algorithm parameters used by these validations specify the top and bottom measurement conditions in a particular measurement condition range. For example, some validations must taken into account whether or not a measurement value was estimated. Estimated measurements are designated through a set of base package measurement conditions including "System Estimate", "External Estimate, and "Office Estimate" that span a numeric range from 201000 through 402000. Specifying a "top range value" of 201000, and a "bottom range value" of 402000, means that any measurement whose measurement condition falls within that range is considered to be estimated.
- **Validation Algorithm Type / Algorithm**: The base package algorithm type and algorithm specified in the "Validation" system event for the rule, if applicable. If specified, a brief description of the algorithm is also provided. Validation algorithms used by VEE rules do not use "soft" parameters.
- **Rule Parameters**: Parameters used to define the rule (if applicable). Does not include common parameters used by all rules (see **Common Parameters** on page 7-19).
- **Processing Logic**: A description of the processing performed by the rule. Processing logic is performed by the "Apply VEE Rule" algorithm specified on the VEE rule business object.
- **Example**: An example configuration of the rule. Note that all examples use default values for all "Apply VEE Rule Algorithm Parameters".

Common Parameters

All VEE rules use the following common parameters:

- **VEE Group**: The VEE group to which the rule belongs
- **VEE Rule**: The user-defined name given to the VEE rule
- **Sequence**: The sequence within its VEE group that the rule will be executed
- **Description**: A short description of the rule. This is used as the Information String for the rule in the user interface.
- **Detailed Description**: A detailed description of the rule
- **Category**: The rule's category. Base package options include "Automatic Correction of Invalid Data," "Estimation Rules," and "Validation Rules."
- **Start Date**: The date on which the rule is considered in effect. VEE rules are not applied to initial measurements whose start dates are earlier than this date.
- **End Date**: The date after which the rule is considered no longer in effect. VEE rules are not applied to initial measurements whose stop dates are later than this date.

Validation Rules

This section provides descriptions for the base package VEE rules used in validation of initial measurement data. Validation rules are used to validate that an initial measurement is within expected tolerances, and is correct. Some validation rules can also perform edits to the initial measurement. Base package validation rules include:

- **Duplicate IMD Check**
- **Ensure IMD Exists for Sibling MCs**
- **High/Low Check**
- **Interval Replacement Rule**
- **Interval Size Validation**
- **Interval Spike Check**
- **Multiplier Check**
- **Negative Consumption Check**
- **Raise Missing Quantity Exception**
- **Scalar Replacement Rule**
- **Sum Check**
- **Unit Of Measure Check**
- **Zero Consumption Check**

Duplicate IMD Check

Duplicate IMD Check rules check to determine whether the current initial measurement is a duplicate of one already received for the same measuring component. If the initial measurement is determined to be a duplicate, the rule produces a VEE exception of the type and severity defined on the rule when it is configured.

- **Rule Name**: Duplicate IMD Check
- **Base Package VEE Rule Business Object**: D1-DuplicateIMDCheck
- **Apply VEE Rule Algorithm Type / Algorithm**: D1-DUPIMDCHK
- **Apply VEE Rule Algorithm Parameters**: N/A
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
    - **Duplicate IMD Exception Configuration**: Defines the Exception Type and Severity for exceptions created when a duplicate IMD is detected. In addition, an Override To Do Type and Override To Do Role can also be defined.
- **Processing Logic**

  Duplicate IMD Check rules check for duplicate initial measurements by looking for:

    - Initial measurements associated to the same measuring component as the current initial measurement
    - Initial measurements that utilize the same business object as the current initial measurement (for example, Initial Load Scalar)
    - Initial measurements that reference the same To Date/Time (ends on the same date) as the current initial measurement
    - Initial measurements that exist in a "Finalized" state

- Initial measurements that with pre-VEE contents identical to the pre-VEE contents of the current initial measurement

  If any existing initial measurements are found that meet all of the above criteria, the current initial measurement is deemed to be a duplicate, and an exception is created using the Exception Type and Severity defined for the "Duplicate IMD Exception Configuration" parameters. This exception will have a Message Category of 11802 and a Message Number of 10021.

- **Example**: The following VEE rule creates an exception if the current initial measurement is determined to be a duplicate of an existing "Final" initial measurement.for the same measuring component.

  VEE Group: Initial Load Validations

  VEE Rule: DUPLICATE_CHECK

  Sequence: 10

  Description: Duplicate IMD Check

  Category: Validation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Duplicate IMD Exception Configuration:

  - **Exception Type**: Duplicate IMD Detected
  - **Exception Severity**: Terminate
  - **Override To Do Type**: N/A
  - **Override To Do Role**: N/A

Ensure IMD Exists for Sibling MCs

Ensure IMD Exists for Sibling MCs rules check to ensure that all sibling measuring components of the current initial measurement's measuring component each have initial measurements associated to them for the same period as the current initial measurement. If one or more of sibling measuring components does not have an initial measurement for the same period as the current initial measurement, the rule creates an exception. Sibling measuring components are those associated to the same device configuration as the current initial measurement's measuring component.

- **Rule Name**: Ensure IMD Exists for Sibling MCs
- **Base Package VEE Rule Business Object**: D2-EnsureIMDExistsForSibling
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-ENSIMDMC
- **Apply VEE Rule Algorithm Parameters**: N/A
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
  - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.
  - **IMD Not Found Exception**: Defines the Exception Type and Severity for exceptions created when no initial measurement data can be found for the sibling measuring component.
- **Processing Logic**

  Ensure IMD Exists for Sibling MCs rules validate that initial measurements exist for all of the other measuring components associated to the same device configuration as the current measuring component, for the same period of time as the initial measurement being validated. These rules also check that all of the initial measurements have the same External Source ID (indicating that they all came from the same usage file).

If either of these validations fails, an exception is created using the Exception Type and Severity defined for the “IMD Not Found Exception” parameter. This exception will have a Message Category of 11802 and a Message Number of 10021.

If the measuring component is not supplied in the initial measurement, an exception is created using the Exception Type and Severity defined for the “Insufficient Input Data Exception” parameter. This exception will have a Message Category of 11802 and a Message Number of 10115.

If an external source id is not supplied in the initial measurement, an exception is created using the Exception Type and Severity defined for the “Insufficient Input Data Exception” parameter. This exception will have a Message Category of 11802 and a Message Number of 10020.

- **Example**: The following VEE rule creates an exception if initial measurements are not found for all sibling measuring components of the current initial measurement's measuring component.

  VEE Group: Initial Load Validations

  VEE Rule: CHECK_SIBLING_MC_IMD

  Sequence: 10

  Description: Check IMDs for Sibling MCs

  Category: Validation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Insufficient Input Data Exception:

  - **Exception Type**: Insufficient Input Data
  - **Exception Severity**: Issues

  IMD Not Found Exception:

  - **Exception Type**: No Data Found
  - **Exception Severity**: Issues

High/Low Check

High/Low Check rules compare consumption of an incoming initial measurement against historical data as a means of assuring reasonable data. High/Low Check rules can be used with both scalar and interval initial measurements.

- **Rule Name**: High/Low Check
- **Base Package VEE Rule Business Object**: D2-VEERuleHighLowCheck
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-HILO-CHK
- **Apply VEE Rule Algorithm Parameters**:
  - Condition - bottom range value representing Non-Normal measurements
  - Condition - top range value representing Non-Normal measurements
  - Condition - bottom range value for System-Estimated measurements
  - Condition - top range value for System-Estimated measurements
  - Condition - bottom range value for measurements during an Outage

    - Condition - top range value for measurements during an Outage

- **Validation Algorithm Type / Algorithm**: D2-HILO-VAL

  This algorithm ensures that if the rule is configured to perform a high check or a low check (or both) that the corresponding tolerance is supplied. For each option, this validation ensures that either a tolerance percentage or a tolerance factor is configured, but not both.

- **Rule Parameters**:
    - **High/Low Check**: Indicates if the rule should check for high usage, low usage, or both when comparing the initial measurement to historical data.
    - **High Tolerance (%)**: The percentage tolerance by which the measurement data can exceed historical data before failing the validation.
    - **High Tolerance Factor**: A factor containing the tolerance by which the measurement data can exceed historical data before failing the validation.
    - **Low Tolerance (%)**: The percentage tolerance by which the measurement data can be below historical data before failing the validation.
    - **Low Tolerance Factor**: A factor containing the tolerance by which the measurement data can be below historical data before failing the validation.
    - **Required Historical Data (%)**: The percentage amount of historical data required for the validation to be executed, relative to the duration of the initial measurement being validated. For example, if the duration of the initial measurement is 1 day and this parameter is set to 50%, the system must find at least ½ day of historical data in order for the rule to execute. Note that this field can be set to values higher than 100%.
    - **Historical Pre-Window**: The number of days prior to the start date to find historical data.
    - **Historical Post-Window**: The number of days after the stop date to find historical data.
    - **Allowed Historical User Edited Data(%)**: The percentage amount of historical data allowed if some of the data has been user edited.
    - **Allowed Historical System Estimated Data(%)**: The percentage amount of historical data allowed if some of the data has been system estimated.
    - **Allowed Historical Non Normal Data(%)**: The percentage amount of historical data allowed if some of the data is "non-normal". This parameter is only used for interval measuring component types.
    - **Historical Look First**: Specifies the first point in time to find historical data. Options include “Last Reading”, “Last Year”, and “Input Dates.”
    - **Comparison Method**: The method by which historical data and the measurement data are compared (Average or Maximum).
    - **Perform Outage Check**: Specifies the rule should check for an outage for the device that supplied the measurement.
    - **Outage Activity Type**: Specifies an activity type that designates an outage for the device that supplied the measurement.
    - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.
    - **No Historical Data Found Exception**: Defines the Exception Type and Severity for exceptions created when there is no historical data found.
    - **High Check Exception**: Defines the Exception Type and Severity for exceptions created when the measurement fails a high check.

- **Low Check Exception**: Defines the Exception Type and Severity for exceptions created when the measurement fails a low check.
- **Low Check Outage Exception**: Defines the Exception Type and Severity for exceptions created when the rule detects an outage for the device that supplied the measurement.

- **Processing Logic:**

High/Low Check rules compare either the Average Daily Usage (ADU) or the maximum value of the initial measurement being validated with the ADU / max for historical measurements. Historical measurements can be based on measurements from either one year ago or from the most recent reading for the initial measurement's measuring component. If insufficient usable data is found using the first historical option, measurement data for the second historical option is evaluated. These rules can be configured to specify whether to check measurements from one year ago or the most recent reading first. It can also specify whether to perform both High and Low checking, or only one or the other.

When performing a High check, the rule checks to determine if the ADU/Maximum of the initial measurement is greater than (within a tolerance defined by the "High Tolerance (%)" and "High Tolerance Factor" parameters) of the historical data. When performing a Low check, the rule checks to determine if the ADU/Maximum of the initial measurement is less than (within a tolerance defined by the "Low Tolerance (%)" and "Low Tolerance Factor" parameters) of the historical data.

The "Required Historical Data" parameter defines the percentage of historical data required for the rule to execute. This percentage is relative to the duration of the initial measurement being validated. If the specified amount of historical data is not found, the rule does not execute.

The Rule can be configured to reject historical data based on the condition codes of the data. These options can be used to find measurement data that is likely to produce a reasonable ADU/Maximum for comparison to the current data.

Non-Normal Data

The "Allowed Historical Non Normal Data(%)" parameter specifies the maximum percentage amount of historical data allowed if some of the data is "non-normal". For interval data, the percentage calculation is calculated based on the number of measurements compared to the total number of measurements in the period. For scalar data, the time span of the measurement relative to the length of the historical period is used to determine the percentage

The "Condition - bottom range value representing Non-Normal measurements" and "Condition - top range value representing Non-Normal measurements" parameters are used to define the range of condition values for measurements that should be categorized as Non-Normal. For example, Non-Normal data might be any measurement whose condition code is between Missing (201000) and Office Estimate (402000). Depending upon how condition codes are configured, the bottom range condition parameter might be set as 200000, and the top range condition parameter as 402999. If custom conditions have been added in the Office Estimate category, such as 402500 (corresponding to "Office Estimate to Correct Spike", for example) measurements referencing those custom conditions will also be treated as "Non-Normal".

System Estimated Data

The "Allowed Historical System Estimated Data(%)" parameter specifies the maximum percentage amount of historical data allowed if some of the data has been system estimated.

The "Condition - bottom range value representing System-Estimated measurements" and "Condition - top range value representing System-Estimated measurements" parameters are used to define the range of condition values for measurements that should be categorized as System-Estimated. For example, System-Estimated data might be any measurement whose condition code is between 300000 (below the System-Estimated value of 301000) and 399999 (just below Office Estimate). To evaluate Office Estimates along with System Estimates, the top range can be extended to include Office Estimate condition values.

If the ADU/Maximum for the current initial measurement is higher or lower than the historical ADU/Maximum, taking into account the tolerances configured on the rule, an exception is logged using the Exception Type and Severity defined for the "High Check Exception" or "Low Check Exception" parameters (respectively).

The rule can also check for outages for the initial measurements' device between the measurement's start and end time. Outages can be determined by either the presence of an outage activity for the measurement's device, or if the measurement contains condition codes that fall within the range of "outage" conditions defined on the algorithm. The "Outage Activity Type" parameter specifies the Activity Type used to designate an outage for the measurement's device. The "Condition - bottom range value for measurements during an Outage" and "Condition - top range value for measurements during an Outage" algorithm parameters define the range of outage conditions for the rule. Outage checks are performed when the initial measurement's ADU/Maximum is lower than the historical ADU/Maximum tolerance. If an outage is detected, an exception is created using the Exception Type and Severity defined for the "Low Check Outage Exception" parameter.

- **Example**: The following VEE rule performs both a High check and Low check against historical data. When gathering historical data for comparison, the rule will use up to 5% user-edited data, up to 20% system estimated data, and up to 10% non-normal data. In addition, if consumption for the current initial measurement is deemed to be "low", the rule will also check for outages.

  VEE Group: Initial Load Validations

  VEE Rule: HIGH-LOW-CHECK

  Sequence: 10

  Description: High/Low Check

  Category: Validation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  High Low Check:

  - **High Low Check**: Both
  - **High Tolerance (%)**: 200
  - **Low Tolerance (%)**: 49
  - **Required Historical Data (%)**: 20
  - **Historical Pre-Window**: 10
  - **Historical Post-Window**: 0
  - **Allowed Historical User Edited Data (%)**: 5
  - **Allowed Historical System Estimated Data (%)**: 20
  - **Allowed Historical Non Normal Data (%)**: 10
  - **Historical First Look**: Last Reading
  - **Comparison Method**: Average
  - **Perform Outage Check**: Perform If Low Consumption

- **Outage Activity Type**: Outage Activity

Insufficient Input Data Exception:

- **Exception Type**: Insufficient Input Data
- **Exception Severity**: Issues

No Historical Data Found Exception:

- **Exception Type**: No Data Found
- **Exception Severity**: Issues

High Check Exception:

- **Exception Type**: Consumption Exceeds Threshold
- **Exception Severity**: Issues

Low Check Exception:

- **Exception Type**: Consumption Less Than Threshold
- **Exception Severity**: Issues

Low Check Outage Exception:

- **Exception Type**: Consumption Less Than Threshold due to Outage
- **Exception Severity**: Issues

Interval Replacement Rule

Interval Replacement rules determine whether or not an interval reading will replace any existing measurements and specify whether or not to allow this to happen.

- **Rule Name**: Interval Replacement Rule
- **Base Package VEE Rule Business Object**: D2-IntervalReplacementRule
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-INTREPRL
- **Apply VEE Rule Algorithm Parameters**: N/A
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
  - **Replacement Handling Method**: Indicates if the rule should reject all replacement readings (Reject All), or reject only those replacement readings that would replace a reading that has been manually edited (Reject If Existing Data Is Manually Edited)
  - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.
  - **Interval Replacement Exception**: Defines the Exception Type and Severity for exceptions created when this rule replaces an existing reading.
- **Processing Logic**:

  Interval Replacement rules check if the intervals in an interval initial measurement will replace existing final measurements.

  If the “Replacement Handling Method” parameters is set to "Reject All", the rule creates an exception and exits from processing if the incoming measurement would replace existing final measurements using the Exception Type and Severity defined for the “Interval Replacement Exception” parameter. This exception will have a Message Category of 11802 and a Message Number of 10007.

  If the “Replacement Handling Method” parameters is set to "Reject If Existing Data Is Manually Edited", the rule creates logs an exception and exits from processing if the incoming measurement would replace existing final measurements that had been user (manually) edited. The exception is created using the Exception Type and Severity defined for the “Interval Replacement Exception” parameter. This exception will have a Message Category of 11802 and a Message Number of 10022.

- **Example**: The following VEE rule determines whether or not an interval reading would replace any existing measurements and if so, will reject the incoming measurements.

  VEE Group: Interval Validations

  VEE Rule: INTERVAL_REPLACEMENT

  Sequence: 10

  Description: Interval Replacement Rule

  Category: Validation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Replacement Handling Method: Reject All

  Insufficient Input Data Exception:

  - **Exception Type**: Insufficient Input Data
  - **Exception Severity**: Issues

  Interval Replacement Exception:

  - **Exception Type**: Interval Data Replacement
  - **Exception Severity**: Issues

Interval Size Validation

Interval Size Validation rules check to ensure that the interval size of the incoming initial measurement data matches the value defined in the measuring component type.

- **Rule Name**: Interval Size Validation
- **Base Package VEE Rule Business Object**: D2-IntervalSizeValidation
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-INTSIZVAL
- **Apply VEE Rule Algorithm Parameters**: N/A
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
  - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.

    Interval Size Validation rules require the interval size (Seconds-Per-Interval or SPI) be supplied in the initial measurement.
  - **Invalid Interval Size Exception**: Defines the Exception Type and Severity for exceptions created if the measurement fails this validation.
- **Processing Logic**:

  Interval Size Validation rules validate that the seconds per interval (SPI) supplied with the initial measurement being validated is equal to the interval size defined on measurement’s measuring component type. If these are not equal, the rule creates an exception using the Exception Type and Severity defined for the “Invalid Interval Size Exception” parameter. This exception will have a Message Category of 11802 and a Message Number of 10009.

If the SPI is not supplied in the initial measurement, an exception is created using the Exception Type and Severity defined for the "Insufficient Input Data Exception" parameter. This exception will have a Message Category of 11802 and a Message Number of 10018.

- **Example**: The following VEE rule issues an exception if the interval size of the incoming initial measurement does not match the value defined for its measuring component type.

  VEE Group: Interval Validations

  VEE Rule: SPI_CHECK

  Sequence: 10

  Description: SPI Check

  Category: Validation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Insufficient Input Data Exception:

  - **Exception Type**: Insufficient Input Data
  - **Exception Severity**: Issues

  Invalid Interval Size Exception:

  - **Exception Type**: Interval Size Discrepancy
  - **Exception Severity**: Issues

Interval Spike Check

Interval Spike Check rules examine interval data to identify intervals with conspicuously high usage relative to surrounding intervals. Interval Spike Check rules identify spikes by subtracting the third-highest peak from the highest peak minus, and dividing the result by the third-highest peak. If the percent result is larger than the tolerance configured on the rule, the rule creates an exception.

- **Rule Name**: Interval Spike Check
- **Base Package VEE Rule Business Object**: D2-IntervalSpikeCheck
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-INTSPKCHK
- **Apply VEE Rule Algorithm Parameters**: N/A
- **Validation Algorithm Type / Algorithm**: D2-INTSPKVLD

  This algorithm ensures that the "minimum number of intervals" parameter on the Interval Spike Check VEE rule has a value greater than or equal to 3. (The rule requires the first- and third-highest intervals, and so by definition requires 3 intervals.)

- **Rule Parameters**:
  - **Minimum Number of Intervals**: The minimum number of interval data values required in the measurement for the rule to be executed.
  - **Spike Check Method**: Specifies whether to check for spikes for the entire interval cut (Entire Interval Cut), or for each 24 hour period within the cut (Repeat for Each 24 Hour Period).
  - **Spike Tolerance (%)**: Percentage value used to determine if an interval data value is considered a spike. This value is compared against a value equal to the highest peak minus the third-highest peak, divided by the third-highest peak. If this value is greater than the specified tolerance, the measurement fails this validation, and an exception is generated.

    - **Condition Flag Value to identify spike(s)**: The condition value assigned to interval data values that are adjusted as a result of being identified as a spike.
    - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.
    - **Interval Spike Detected**: Defines the Exception Type and Severity for exceptions created when a spike is identified within the measurement.

- **Processing Logic**:

  Interval Spike Check rules identify spikes by subtracting the third-highest peak from the highest peak, and dividing the result by the third-highest peak. If the percent result is larger than the "Spike Tolerance (%)" parameter, the rule creates an exception using the Exception Type and Severity defined for the "Interval Spike Detected" parameter.

  These rules can be executed in one of two modes, based on the "Spike Check Method" parameter:

  - Repeat for Each 24 Hour Period: The spike check is performed for every 24-hour portion of data in the initial measurement.
  - Entire Interval Cut: The spike check is performed for the entire initial measurement

  Note that this rule uses a measurement service to identify spikes. To access this logic outside of the context of this rule, please refer to the D1-IdentifySpikes business service along with its underlying service.

- **Example**: The following VEE rule checks the entire measurement for interval spikes, and flags those interval values identified as spikes, based on a spike tolerance of 70%.

  VEE Group: Interval Validations

  VEE Rule: SPIKE_CHECK

  Sequence: 10

  Description: Interval Spike Check

  Category: Validation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Minimum Number of Intervals: 8

  Spike Check Method: Entire Cut

  Spike Tolerance (%): 70

  Condition Flag Value to Interval Spike(s): Spike - Treat As Missing

  Insufficient Input Data Exception:

  - **Exception Type**: Insufficient Input Data
  - **Exception Severity**: Issues

  Interval Spike Check Exception:

  - **Exception Type**: Interval Check Detected
  - **Exception Severity**: Issues

Multiplier Check

Multiplier Check rules check to ensure that the device multiplier value of the incoming initial measurement data matches the multiplier value stored on the measuring component.

- **Rule Name**: Multiplier Check
- **Base Package VEE Rule Business Object**: D2-RegisterMultiplierCheck
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-REGMULCHK
- **Apply VEE Rule Algorithm Parameters**: N/A
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
  - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.

    Multiplier Check rules require the register multiplier be supplied in the initial measurement.
  - **Multiplier Exception**: Defines the Exception Type and Severity for exceptions created if the measurement fails this validation
- **Processing Logic**:

  Multiplier Check rules validate that the register multiplier supplied with the initial measurement is equal to the multiplier stored on the measuring component. If these are not equal, the rule creates an exception using the Exception Type and Severity defined for the "Multiplier Exception" parameter. This exception will have a Message Category of 11802 and a Message Number of 10010

  If the register multiplier is not supplied in the initial measurement, the rule creates an exception using the Exception Type and Severity defined for the "Insufficient Input Data Exception" parameter. This exception will have a Message Category of 11802 and a Message Number of 10015.
- **Example**: The following VEE rule checks if the meter multiplier supplied in the initial measurement differs from the meter multiplier defined for the measuring component.

  VEE Group: Initial Load Validations

  VEE Rule: MULTIPLIER_CHECK

  Sequence: 10

  Description: Multiplier Check

  Category: Validation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Insufficient Input Data Exception:

  - **Exception Type**: Insufficient Input Data
  - **Exception Severity**: Issues

  Multiplier Exception:

  - **Exception Type**: Device Multiplier Discrepancy
  - **Exception Severity**: Issues

Negative Consumption Check

Negative Consumption Check rules check for negative consumption. Negative consumption occurs if the total consumption for the initial measurement (calculated by summing all measurements from the initial measurement) is less than zero. Note that Negative Consumption Check rules should not be applied to measurements received from devices that can have negative readings.

- **Rule Name**: Negative Consumption Check
- **Base Package VEE Rule Business Object**: D2-NegativeConsumptionCheck
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-NCON-CHK
- **Apply VEE Rule Algorithm Parameters**: N/A
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
  - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.
  - **Negative Consumption**: Defines the Exception Type and Severity for exceptions created when negative consumption is detected.
- **Processing Logic**:

  Negative Consumption Check rules create an exception if the total consumption, calculated by summing all measurements from the initial measurement, is less than zero. The exception is created using the Exception Type and Severity defined for the “Negative Consumption” parameters. Note if the rule encounters negative consumption, an exception will be created only if the measuring component type is **not** configured to "Allow Negative Consumption".
- **Example**: The following VEE rule checks for negative consumption and creates an exception if negative consumption is detected.

  VEE Group: Initial Load Validations

  VEE Rule: NEGATIVE_CONSUMPTION_CHECK

  Sequence: 10

  Description: Negative Consumption Check

  Category: Validation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Insufficient Input Data Exception:

  - **Exception Type**: Insufficient Input Data
  - **Exception Severity**: Issues

  Negative Consumption Exception:

  - **Exception Type**: Unexpected Negative Consumption
  - **Exception Severity**: Issues

Raise Missing Quantity Exception

Raise Missing Quantity Exception rules create an exception if a specified percentage of measurements within the initial measurement are missing, excluding intervals associated with an Outage event.

- **Rule Name**: Raise Missing Quantity Exception
- **Base Package VEE Rule Business Object**: D2-RaiseMissingQuantityExcp

- **Apply VEE Rule Algorithm Type / Algorithm**: D2-RAIMISQTY
- **Apply VEE Rule Algorithm Parameters**:
  - Condition - bottom range value to include
  - Condition - top range value to include
  - Condition - bottom range value to exclude
  - Condition - top range value to exclude
  - Threshold Percentage XPath
  - Exception Type XPath
  - Exception Severity XPath
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
  - **Percentage Threshold of Missing Intervals**: Defines the percentage of missing intervals that must fall within the "missing" range before an exception will be created. The "missing" range is defined on the algorithm used by this rule. Note that this parameter is not required when validation usage for scalar measuring components.
  - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.
  - **Missing Quantity Exception**: Defines the Exception Type and Severity for exceptions created when the percentage of "missing" interval measurements exceeds the Percentage Threshold of Missing Intervals.
- **Processing Logic**:

  Raise Missing Quantity Exception rules create an exception if the percentage of measurements within an interval initial measurement (defined by the “Percentage Threshold of Missing Intervals” parameter) is considered “missing.” The exception is created using the Exception Type and Severity defined for the “Missing Quantity Exception” parameter.

  Missing intervals are those marked with a condition code that falls between the condition range defined by the “Condition - bottom range value to include” and the “Condition - top range value to include” algorithm parameters. The “Condition - bottom range value to exclude” and the “Condition - top range value to exclude” algorithm parameters can be used to define a range of conditions for intervals to be excluded by the rule. The condition ranges can overlap (the range for “missing” might contain a superset of conditions that includes those for “outage” or some other condition). For example, assume the condition range for "missing" spans entirely the condition range for "outage". This rule could be configured to raise an exception for any interval that has a value in the range of "missing" conditions, but at the same time exclude intervals marked with "outage" conditions from triggering the exception creation.

  The “Threshold Percentage XPath”, “Exception Type XPath” and “Exception Severity XPath” algorithm parameters identify the XPath location within the VEE rule BO schema for the corresponding elements. These parameters should only be changed if a custom version of this rule is created, and the XPath location for these elements in the VEE rule BO schema is changed from the BO schema provided with the base package.
- **Example**: The following VEE rule creates an exception if more than 10 percent of the interval values within an initial measurement are missing.

  VEE Group: Interval Validations

  VEE Rule: MISSING_EXCEPTION

  Sequence: 10

Description: Excessive Gaps

Category: Validation Rules.

Start Date: 05-01-2010

End Date: N/A

Missing Quantity Data:

- **Percentage Threshold of Missing Intervals**: 10

Insufficient Input Data Exception:

- **Exception Type**: Insufficient Input Data
- **Exception Severity**: Issues

Missing Quantity Exception:

- **Exception Type**: Gaps Detected in Data
- **Exception Severity**: Issues

Scalar Replacement Rule

Scalar Replacement rules identify whether or not a scalar reading will completely replace an existing measurement and specify whether or not to allow this to happen.

- **Rule Name**: Scalar Replacement Rule
- **Base Package VEE Rule Business Object**: D2-ScalarReplacementRule
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-SCAREPRL
- **Apply VEE Rule Algorithm Parameters**: N/A
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
    - **Replacement Handling Method**: Indicates if the rule should reject all replacement readings (Reject All), or reject only those replacement readings that would replace a reading that has been manually edited (Reject If Existing Data Is Manually Edited)
    - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.
    - **Scalar Replacement Exception**: Defines the Exception Type and Severity for exceptions created when this rule replaces an existing reading.
- **Processing Logic**:

    Scalar Replacement rules check if a scalar initial measurement will replace an existing final measurement.

    If the “Replacement Handling Method” parameters is set to "Reject All", the rule creates an exception and exits from processing if the incoming measurement would replace an existing final measurement using the Exception Type and Severity defined for the “Scalar Replacement Exception” parameter. This exception will have a Message Category of 11802 and a Message Number of 10008.

    If the “Replacement Handling Method” parameters is set to "Reject If Existing Data Is Manually Edited", the rule creates logs an exception and exits from processing if the incoming measurement would replace an existing final measurement that had been user (manually) edited. The exception is created using the Exception Type and Severity defined for the “Scalar Replacement Exception” parameter. This exception will have a Message Category of 11802 and a Message Number of 10023.

- **Example**: The following VEE rule determines whether or not a scalar reading will replace existing measurements and if so, whether it will reject the incoming measurements if the existing measurements were manually edited.

  VEE Group: Scalar Validations

  VEE Rule: SCALAR_REPLACEMENT

  Sequence: 10

  Description: Scalar Replacement

  Category: Validation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Replacement Handling Method: Reject If Existing Data Is Manually Edited

  Insufficient Input Data Exception:

  - **Exception Type**: Insufficient Input Data
  - **Exception Severity**: Issues

  Scalar Replacement Exception:

  - **Exception Type**: Scalar Data Replacement
  - **Exception Severity**: Issues

Sum Check

Sum Check rules compare the difference between the total consumption of an incoming initial measurement against the total consumption (based on final measurements) for a related measuring component on the same device over the same time period. If the difference exceeds a specified tolerance, the rule creates an exception. Related measuring components are those defined as “Consumption Reference Measuring Components” for the initial measurement’s measuring component.

- **Rule Name**: Sum Check
- **Base Package VEE Rule Business Object**: D2-SumCheck
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-SUM-CHK
- **Apply VEE Rule Algorithm Parameters**: N/A
- **Validation Algorithm Type / Algorithm**: D2-SUMCHKVAL

  This algorithm ensures that for the Sum Check VEE rule, only one sum check tolerance type is supplied (either percentage, hard value, or meter multiplier-based), and that its value is not negative.

- **Rule Parameters**:
  - **Sum Check**: Defines the different types of tolerance that can be used by this rule, including:
    - **Percentage Tolerance (%)**: The percentage difference between the consumption of the two measuring components. If the percentage difference is higher than this tolerance, the measurement fails the validation.
    - **Tolerance**: The difference between the consumption of the two measuring components. If the difference is higher than this tolerance, the measurement fails the validation.
    - **Meter Multiplier Tolerance (%)**: The tolerance based on the multiples of the meter multiplier. For example, if this tolerance is set to "2", for a measuring

component that measures KWH with a multiplier 3.5, an exception is created if the difference between the consumption and the total related consumption exceeds 7% (3.5*2). If the percentage difference is higher than this tolerance, the measurement fails the validation.

Note: Only one tolerance can be defined for this rule.

- **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.
- **Sum Check**: Defines the Exception Type and Severity for exceptions created when consumption does not match between the initial measurement data and measurement data for the related measuring component.
- **No Data Found**: Defines the Exception Type and Severity for exceptions created when there is no data found for related measuring components.

- **Processing Logic**:

  Sum Check rules evaluate whether consumption for the current initial measurement is within a specified tolerance of the sum of the consumption during the same time period for any measuring components related to the current initial measurement's measuring component. If the values are not within the defined tolerance of each other, an exception is created using the Exception Type and Severity defined for the "Sum Check Exception" parameter. If no measurement data can be found for the related measuring component for the same time period, an exception is created using the Exception Type and Severity defined for the "No Data Found Exception" parameter.

  The rule can be used to evaluate consumption totals for an interval measuring component that has a related scalar measuring component with the same UOM, to ensure that the total consumption of the interval measuring component is within a tolerance of that of the scalar value. It can also be used to evaluate consumption totals for scalar TOU meters that have a "check" register (for example, three registers that measure ON-PEAK, OFF-PEAK, and SHOULDER, with a fourth check register that measures the total consumption).

- **Example**: The following VEE rule creates an exception if the percentage difference between the consumption of the incoming initial measurement and consumption for its related measuring components is more than 95%.

  VEE Group: Interval Validations

  VEE Rule: SUM_CHECK

  Sequence: 10

  Description: Sum Check

  Category: Validation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Sum Check:

  - **Percentage Tolerance (%)**: 95
  - **Tolerance**: N/A
  - **Meter Multiplier Tolerance**: N/A

  Insufficient Input Data Exception:

  - **Exception Type**: Insufficient Input Data
  - **Exception Severity**: Issues

  Sum Check Exception:

- **Exception Type**: Consumption Differs from Reference Amount
- **Exception Severity**: Issues

No Data Found Exception:

- **Exception Type**: No Data Found
- **Exception Severity**: Issues

Unit Of Measure Check

Unit Of Measure Check rules check that the Unit of Measure (UOM) of the incoming initial measurement data matches the UOM specified on the measuring component's type.

- **Rule Name**: Unit of Measure Check
- **Base Package VEE Rule Business Object**: D2-UOMCheck
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-UOMCHK
- **Apply VEE Rule Algorithm Parameters**: N/A
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
    - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.

      Unit of Measure Check rules require that the unit of measure be supplied in the initial measurement.
    - **UOM Check**: Defines the Exception Type and Severity for exceptions created if the measurement fails this validation.
- **Processing Logic**:

  Unit Of Measure Check rules check the unit of measure passed in with the initial measurement against the primary unit of measure configured on the initial measurement's measuring component type. If the two are not equal, the rule creates an exception using the Exception Type and Severity defined for the "Unit Of Measure Exception" parameter. This exception will have a Message Category of 11802 and a Message Number of 10011.

  If the UOM is not supplied on the initial measurement, the rule creates an exception using the Exception Type and Severity defined for the "Insufficient Input Data Exception" parameter. This exception will have a Message Category of 11802 and a Message Number of 10016.
- **Example**: The following VEE rule creates an exception if the unit of measure specified in the initial measurement does not match the unit of measure defined for its measuring components' type.

  VEE Group: Initial Load Validations

  VEE Rule: UOM_CHECK

  Sequence: 10

  Description: Unit of Measure Check

  Category: Validation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Insufficient Input Data Exception:

    - **Exception Type**: Insufficient Input Data

- **Exception Severity**: Issues

Unit of Measure Exception:

- **Exception Type**: Unit of Measure Discrepancy
- **Exception Severity**: Issues

Zero Consumption Check

Zero Consumption Check rules check for zero consumption for an incoming initial measurement, and can also optionally check for an outage (of a specified Outage Activity Type) for the usage period if zero consumption is detected.

- **Rule Name**: Zero Consumption Check
- **Base Package VEE Rule Business Object**: D2-ZeroConsumptionCheck
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-ZEROCNCHK
- **Apply VEE Rule Algorithm Parameters**:
  - Outage Bottom Range Condition
  - Outage Top Range Condition
- **Validation Algorithm Type / Algorithm**: D2-OACHKVAL

  This algorithm validates to ensure that Outage Activity Type is populated if "Perform Outage Check if Zero Consumption" flag is set to “Yes”.
- **Rule Parameters**:
  - **Perform Outage Check If Zero Consumption**: Specifies whether or not the rule should check for an outage
  - **Outage Activity Type**: Specifies the activity type that represents an outage (used if “Perform Outage Check If Zero Consumption” is set to “Yes.”)
  - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.
  - **Zero Consumption**: Defines the Exception Type and Severity for exceptions created when the rule detects zero consumption for the current initial measurement.
- **Processing Logic**:

  Zero Consumption Check rules detect if the total consumption for the current initial measurement is zero. For interval measurements, the rule sums all the Post-VEE measurement quantities to obtain the total consumption for the interval measurement. For scalar measurements, the rule retrieves the Post-VEE measurement quantity to obtain the total consumption. Note: if the current scalar initial measurement is the first measurement for the measuring component (measurement quantity is zero and start date time is not populated), the rule exits.

  If the consumption for the initial measurement is zero and the "Perform Outage Check if Zero Consumption" parameter is set to “Yes”, the rule also checks for outages for the initial measurements’ device between the measurement’s start and end time. Outages can be determined by either the presence of an outage activity for the measurement’s device, or if the measurement contains condition codes that fall within the range of “outage” conditions defined on the algorithm. The “Outage Activity Type” parameter specifies the Activity Type used to designate an outage for the measurement’s device. The “Outage Bottom Range Condition” and “Outage Top Range Condition” algorithm parameters define the range of outage conditions for the rule. If an outage is detected, an exception is created using the Exception Type and Severity defined for the “Zero Consumption Outage” parameter. This exception will have a Message Category of 11802 and a Message Number of 10110.

If the consumption for the initial measurement is zero and an outage is not detected, or if the "Perform Outage Check if Zero Consumption" parameter is set to “No”, the rule creates an exception using the Exception Type and Severity defined for the “Zero Consumption” parameter. This exception will have a Message Category of 11802 and a Message Number of 10025.

- **Example**: The following VEE rule creates an exception if the total consumption for the incoming initial measurement is zero. If the consumption is zero, this rule will also check for outage activities.

  VEE Group: Initial Load Validations

  VEE Rule: ZERO_CONSUMPTION_CHECK

  Sequence: 10

  Description: Zero Consumption Check

  Category: Validation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Perform Outage Check If Zero Consumption: Yes

  Outage Activity Type: Outage Activity

  Insufficient Input Data Exception:

  - **Exception Type**: Insufficient Input Data
  - **Exception Severity**: Issues

  Zero Consumption:

  - **Exception Type**: Zero Consumption
  - **Exception Severity**: Issues

  Zero Consumption Outage:

  - **Exception Type**: Zero Consumption Due to Outage
  - **Exception Severity**: Issues

Estimation Rules

This section provides descriptions for the base package VEE rules used in estimation of initial measurement data. Estimation rules are used to estimate initial measurement data based on existing data. Estimation calculations can be based on averaging of historical data, profile application, or adjusting and calculating values based on related measurement data. Base package estimation rules include:

- **Interval Adjustment From Scalar**
- **Interval Averaging Estimation**
- **Interval Interpolation Estimation**
- **Interval Profile Estimation**
- **Scalar Calculation From Interval**
-
- **Scalar Profile Estimation**

Interval Adjustment From Scalar

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

This adjustment can be performed on either all intervals in the interval measurement, or only on those intervals that fall within a specified condition range. Both options require that a scalar measuring component be related to the current interval measuring component as a "Consumption Reference Measuring Component", and that one or more final measurements be present for the related measuring component between the start and end date/times of the current initial measurement.

- **Rule Name**: Interval Adjustment From Scalar
- **Base Package VEE Rule Business Object**: D2-IntervalAdjustmentFrmScalar
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-INTADJSCA
- **Apply VEE Rule Algorithm Parameters**: N/A
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
    - **Interval Adjustment From Scalar**: Defines the parameters used by the VEE rule, including:
        - **Intervals to Adjust**: Specifies which interval data values should be subject to adjustment. Can be either the entire range of interval data values (All), or only those interval data values that have a condition within a defined range (Restrict by Condition).
        - **Bottom Range Condition Value**: Specifies the bottom of the range of condition values for intervals that will be subject to adjustment.
        - **Top Range Condition Value**: Specifies the top of the range of condition values for intervals that will be subject to adjustment.
        - **Condition Value for Adjusted Intervals**: The condition value assigned to intervals adjusted by this rule.
    - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.
    - **Interval Adjustment Exception**: Defines the Exception Type and Severity for exceptions created to indicate that the measurement has been adjusted.

- **Processing Logic**:

  Interval Adjustment From Scalar rules adjust an interval initial measurement such that the total of the interval values in the measurement equal a scalar value. This rule can adjust either all the interval in the measurement or a subset of intervals, based on condition.

  When the “Intervals to Adjust” parameter is set to “All”, the scalar consumption provides a value that is then used to proportionally adjust all of the intervals in the measurement. The formula used to calculate the value of each interval is:

  (Scalar Consumption / Total Initial Measurement Consumption) * Interval Amount

  If the total of all of the intervals is equal to zero, the rule adjusts all of the intervals to the same value.

  When the “Intervals to Adjust” parameter is set to “Restrict by Condition”, the rule only adjusts those intervals with a condition code that falls within the range of conditions defined by the “Bottom Range Condition Value” and the “Top Range Condition Value” parameters. In this case, the formula used to calculate the value of each interval is:

  (Scalar Consumption / Conditional Initial Measurement Consumption) * Interval Amount

  If the total of the intervals that match the specified condition is equal to zero, the rule adjusts all of the intervals to the same value.

  Any adjusted intervals are assigned a condition based on the “Condition Value for Adjusted Intervals” parameter.

- **Example**: The following VEE rule adjusts all interval values in an interval measurement based on final measurements for a related scalar measuring component, and flags all adjusted intervals as “system estimate”.

  VEE Group: Interval Estimations

  VEE Rule: INTERVAL_ADJUST

  Sequence: 10

  Description: Interval Adjustment From Scalar

  Category: Estimation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Interval Adjustment From Scalar:

  - **Interval to Adjust**: All
  - **Bottom Range Condition Value**:
  - **Top Range Condition Value**:
  - **Condition Value for Adjusted Intervals**: System Estimate

  Insufficient Input Data Exception:

  - **Exception Type**: Insufficient Input Data
  - **Exception Severity**: Information

  Interval Adjustment Exception:

  - **Exception Type**: Interval Measurement Adjusted Per Scalar Value
  - **Exception Severity**: Information

Interval Averaging Estimation

Interval Averaging Estimation rules estimate missing interval values based on averaging of historical data for the same device and measuring component. Rules of this type can be configured to specify a maximum allowed percentage of missing interval values, whether or not to include user-edited measurements when averaging historical data, and other parameters related to selecting historical measurements for averaging.

- **Rule Name**: Interval Averaging Estimation
- **Base Package VEE Rule Business Object**: D2-IntervalAveragingEstimation
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-INTAVGEST
- **Apply VEE Rule Algorithm Parameters**:
  - Missing Bottom Range Condition
  - Missing Top Range Condition
  - Outage Bottom Range Condition
  - Outage Top Range Condition
  - Regular Bottom Range Condition
  - Regular Top Range Condition
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
  - **Interval Averaging Estimation**: Defines the parameters used when estimating missing interval data values using averaging, including:
    - **Work Calendar**: The work calendar used to define which days are work days, weekends, and holidays.
    - **Maximum Percentage Missing Intervals**: The maximum percentage of missing values allowable in the initial measurement being estimated.
    - **Estimate if Not Attached to SP**: Specifies whether or not to estimate missing values if the measurement is not attached to a service point.
    - **Include User Edited Intervals**: Indicates that user edited intervals are included when summing the interval consumption amounts to average.
    - **Holiday Scan Range**: This identifies how many previous/next holidays to use in collecting consumption amounts used for interval estimation.
    - **Sunday Scan Range**: This identifies how many previous/next Sundays to use in collecting consumption amounts used for interval estimation.
    - **Same Day of Week Scan Range**: The number of weeks to check back/forward for intervals on the same day of the week. After this number of weeks has been reached, the rule will switch to looking at ""Like Days"", which scans back/forward on days immediately next to the interval date.
    - **Neighboring Day Scan Range**: The number of days to check back/forward immediately next to the interval date.
    - **Intervals to Average**: The number of days to average when using averages to estimate missing data.
    - **Condition Value for Estimates Created**: The condition value assigned to estimated interval data values created by this rule.
  - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.

- **Max Percentage Missing Intervals Exception**: Defines the Exception Type and Severity for exceptions created if the measurement has more missing interval data values that specified in the **Maximum Percentage Missing Intervals** parameter.

- **Processing Logic**:

  Interval Averaging Estimation rules estimate missing consumption by aggregating consumption history for a measuring component and using the average consumption as the estimated amount. The selection of historical data is based on the type of day being estimated. For example, if the missing interval values fall on a holiday, only historical data from holidays and Sundays are used. Likewise, if missing interval is a non-holiday weekend day, historical data from weekend days is used.

  The number of days to scan, the specific calendar to use, and the number of historical days of data to average are defined as rule parameters.

  The rule calculates the percentage of missing intervals by totaling the number of intervals whose condition codes indicate they are either missing or occurred during an outage. Missing intervals are those marked with a condition code that falls between the condition range defined by the "Missing Bottom Range Condition" and the "Missing Top Range Condition" algorithm parameters. Intervals considered to be missing due to an outage are those marked with a condition code that falls between the condition range defined by the "Outage Bottom Range Condition" and the "Outage Top Range Condition" algorithm parameters.

  The rule calculates average interval values based on "regular" intervals values from historical data. "Regular" intervals are those marked with a condition code that falls between the condition range defined by the "Regular Bottom Range Condition" and the "Regular Top Range Condition" algorithm parameters.

  The algorithm performs estimation as follows:

  First, the rule determines if the measurement is eligible for estimation. Estimation is performed only if the following conditions are true.

  - The measuring component is interval.
  - The measuring component is linked to a service point, or the "Estimate If Not Attached to SP" parameter is set to "Estimate".
  - The percentage of missing intervals is less than the value set for the "Maximum Percentage Missing Intervals" parameter. If the percentage of missing interval is greater than this parameter, the rule creates an exception using the Exception Type and Severity defined for the "Max Percentage Missing Intervals Exception" parameter.

  If the measurement is eligible for estimation, each missing interval is estimated using averaging as follows:

  - For Holiday Estimation:

    Sum all of the measuring component's consumption alternating between previous and next holiday dates, whichever holiday date is nearer to the date/time of the interval to be estimated. This is repeated until either the number of days specified in the "Holiday Scan Range" parameter is reached or until a number of intervals equal to the "Number of Intervals to Average" parameter have been found.

    If the number of days specified in the "Holiday Scan Range" parameter is reached but the number of intervals equal to the "Number of Intervals to Average" parameter is not met, sum all of the measuring component's consumption alternating between previous and next Sundays. This is repeated until either the number of days specified in the "Sunday Scan Range" parameter is reached or until a number of intervals equal to the "Number of Intervals to Average" parameter have been found.

  - For Non-Holiday Estimation:

Sum all of the measuring component's consumption alternating between previous and next same days of the week, excluding holidays. This is repeated until either the number of days specified in the "Same Day Scan Range" parameter is reached or until a number of intervals equal to the "Number of Intervals to Average" parameter have been found.

If the number of days specified in the "Same Day Scan Range" parameter is reached but the number of intervals equal to the "Number of Intervals to Average" parameter is not met, sum all of the measuring component's consumption alternating between previous and next neighboring days excluding holidays until either the number of days specified in the "Neighboring Scan Range" parameter is reached or until a number of intervals equal to the "Number of Intervals to Average" parameter have been found.

- Once a number of "regular" intervals equal to the "Number of Intervals to Average" parameter have been found, estimated consumption is calculated as follows:

  Total Accumulated Consumption / Total Number of Intervals.

- **Example**: The following VEE rule estimates interval values in an initial measurement based on averaging of three historical interval values providing that no more than 10% of the interval values in the measurement are missing.

  VEE Group: Interval Estimations

  VEE Rule: INTERVAL_AVERAGING_ESTIMATION

  Sequence: 10

  Description: Interval Averaging Estimation

  Category: Estimation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Interval Averaging Estimation:

  - **Work Calendar**: US Work Calendar 1
  - **Maximum Percentage Missing Intervals**: 10
  - **Estimate if Not Attached to SP**: Estimate
  - **Include User Edited Intervals**: Include
  - **Same Day of Week Scan Range**: 3
  - **Neighboring Day Scan Range**: 3
  - **Sunday Scan Range**: 3
  - **Holiday Scan Range**: 2
  - **Intervals to Average**: 3
  - **Condition Value Estimates Created**: System Estimate

  Insufficient Input Data Exception:

  - **Exception Type**: Insufficient Input Data
  - **Exception Severity**: Information

  Max Percentage Missing Intervals Exception:

  - **Exception Type**: Maximum Percentage of Missing Data Exceeded
  - **Exception Severity**: Information

Interval Interpolation Estimation

Interval Interpolation Estimation rules estimate gaps of missing interval values based on linear interpolation. When performing interpolation, interval values to either side of the missing intervals are used as the basis for interpolation. An "estimation adder" value equal to (preceding value - subsequent value)/(number of intervals in gap + 1) is added to the interval preceding the first missing interval to calculate the value of the first interval. The estimation adder is then added to the first missing interval's interpolated value to derive the second missing interval value, and process is repeated for each subsequent missing interval.

If missing intervals lie at the beginning or end of the initial measurement, the rule uses final measurements immediately before or after the measurement (respectively) in an attempt to find two reference measurement values for interpolation. If a valid measurement can be found for only one side of a gap, the rule assigns each interval in the gap the value of the available measurement. This is referred to as applying a "flat load".

Interval Interpolation Estimation rules can be configured to specify to specify a maximum duration allowed for interpolation, and a maximum allowed percentage of missing interval values.

- **Rule Name**: Interval Interpolation Estimation
- **Base Package VEE Rule Business Object**: D2-IntervalInterpolationEst
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-INTINTEST
- **Apply VEE Rule Algorithm Parameters**:
  - Condition - bottom range value for Missing
  - Condition - top range value for Missing
  - Condition - bottom range value for Outage
  - Condition - top range value for Outage
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
  - **Interval Interpolation Estimation**: Defines the parameters used when estimating missing interval data values using interpolation, including:
    - **Maximum Percentage Missing Intervals**: The maximum percentage of missing values allowable for estimations.
    - **Estimate if Not Attached to SP**: Specifies whether or not to estimate missing values if the measurement is not attached to a service point.
    - **Maximum Hours to Interpolate**: The maximum number of hours to be interpolated
    - **Condition Value for Estimates Created**: The condition value assigned to estimated interval data values created by this rule.
  - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.
  - **Maximum Hours Exceeded Exception**: Defines the Exception Type and Severity for exceptions created if the measurement has more missing hours of data than the **Maximum Hours to Interpolate** parameter.
  - **Max Percentage Missing Intervals Exception**: Defines the Exception Type and Severity for exceptions created if the measurement has more missing interval data values than specified in the **Maximum Percentage Missing Intervals** parameter.
- **Processing Logic**:

  Interval Interpolation Estimation rules estimate gaps in interval initial measurements using prior and subsequent intervals as starting points for linear interpolation. The rule estimates intervals that are designated as "missing" based on condition codes, but not those designated as having occurred during an outage.

Two pairs of algorithm parameters are used to define the range of condition values that correspond to "missing" and "outage" intervals. The "Condition - bottom range value for Missing" and the "Condition - top range value for Missing" parameters define the condition range for "missing" intervals, while the "Condition - bottom range value for Outage" and the "Condition - top range value for Outage" parameters define the condition range for "outage" intervals. The condition ranges can overlap (the range for "missing" might contain a superset of conditions that includes those for "outage" or some other condition). For example, assume the condition range for "missing" spans entirely the condition range for "outage". This rule could be configured to perform estimation for any interval that has a value in the range of "missing" conditions, but at the same time exclude intervals marked with "outage" conditions from estimation calculation.

The "Maximum Hours to Interpolate" parameter specifies the maximum number of consecutive intervals within a gap before that gap can no longer be interpolated.

When estimating values for intervals in the middle of the initial measurement, interval values to either side of the missing intervals are used as the basis for interpolation. An "estimation adder" value equal to (preceding value - subsequent value)/(number of intervals in gap + 1) is added to the interval preceding the first missing interval to calculate the value of the first interval. The estimation adder is then added to the first missing interval's interpolated value to derive the second missing interval value, and process is repeated for each subsequent missing interval.

If missing intervals lie at the beginning or end of the initial measurement, the rule uses final measurements immediately before or after the measurement (respectively) in an attempt to find two reference measurement values for interpolation. If a valid measurement can be found for only one side of a gap, the rule assigns each interval in the gap the value of the available measurement. This is referred to as applying a "flat load". Note that final measurements used for interpolation at the beginning or end of an initial measurement must not be either "missing" or "outage" intervals, based on the algorithm parameters described above. In the event that the gap to be estimated is the entire length of the initial measurement (and the rule is configured such that this is not too large of a gap), the rule attempts to find final measurements as described above.

Estimated intervals are assigned a condition value as defined for the "Condition Value for Estimates Created" parameter.

- **Example**: The following VEE rule estimates up to three hours of missing interval values using linear interpolation providing that no more than 20% of the interval values in the measurement are missing.

  VEE Group: Interval Estimations

  VEE Rule: INTERPOLATE_GAPS

  Sequence: 10

  Description: Interpolate Gaps

  Category: Estimation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Interval Interpolation Estimation:

  - **Maximum Percentage Missing Intervals**: 20
  - **Estimate if Not Attached to SP**: Estimate
  - **Maximum Hours to Interpolate**: 3

- **Condition Value for Estimates Created**: System Estimate

Insufficient Input Data Exception:

- **Exception Type**: Insufficient Input Data
- **Exception Severity**: Information

Maximum Hours Exceeded Exception:

- **Exception Type**: Maximum Gaps Size for Estimation Exceeded
- **Exception Severity**: Information

Max Percentage Missing Intervals Exception:

- **Exception Type**: Maximum Percentage of Missing Data Exceeded
- **Exception Severity**: Information

Interval Profile Estimation

Interval Profile Estimation rules estimate missing interval values based on an associated profile measuring component. Rules of this type can be configured to specify to specify a maximum allowed percentage of missing interval values. Note that the profile measuring component must contain measurement data for the same time period as the initial measurement to be estimated.

- **Rule Name**: Interval Profile Estimation
- **Base Package VEE Rule Business Object**: D2-IntervalProfileEstimation
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-INTPROEST
- **Apply VEE Rule Algorithm Parameters**:
    - Condition - bottom range value for Missing
    - Condition - top range value for Missing
    - Condition - bottom range value for Outage
    - Condition - top range value for Outage
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
    - **Interval Profile Estimation**: Defines the parameters used when estimating missing interval data values from a profile, including:
        - **Maximum Percentage Missing Intervals**: The maximum percentage of missing values allowable for estimations.
        - **Estimate if Not Attached to SP**: Specifies whether or not to estimate missing values if the measurement is not attached to a service point.
        - **Profile Factor**: The profile factor to be used when estimating interval data values.
        - **Condition Value for Estimates Created**: The condition value assigned to estimated interval data values created by this rule.
    - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.
    - **Max Percentage Missing Intervals Exception**: Defines the Exception Type and Severity for exceptions created if the measurement has more missing interval data values than specified in the **Maximum Percentage Missing Intervals** parameter.
- **Processing Logic**:

  Interval Profile Estimation rules use a profile measuring component's interval consumption as a source of values to assign to intervals in the current initial measurement that are designated "missing" (based on the interval's condition). The rule does not estimate intervals designated as an "outage".

Two pairs of algorithm parameters are used to define the range of condition values that correspond to "missing" and "outage" intervals. The “Condition - bottom range value for Missing” and the “Condition - top range value for Missing” parameters define the condition range for “missing” intervals, while the “Condition - bottom range value for Outage” and the “Condition - top range value for Outage” parameters define the condition range for “outage” intervals.

For each interval in the current initial measurement that falls into the category of "missing" and does not fall into the category of "outage", the measurement value for the same date and time from the profile measuring component (defined by the “Profile Factor” parameter) is used as the estimated value. The condition of the estimated interval is then updated to a new condition defined by the “Condition Value for Estimates Created” parameter.

If a measurement is not available for the profile measuring component on the date and time of an interval, the interval is left unchanged.

This rule will only estimate a specified percentage of missing intervals. This percentage is defined by the “Maximum Percentage Missing Intervals”. If the initial measurement has more missing intervals than this percentage, the rule creates an exception using the Exception Type and Severity defined for the “Max Percentage Missing Exception” parameter. If consumption data for the profile measuring component cannot be found for the date and time of an interval to be estimated, the rule creates an exception using the Exception Type and Severity defined for the “Insufficient Input Data Exception” parameter.

- **Example**: The following VEE rule estimates interval measurements based on the “kWh - Profile:” profile measuring component (defined as a factor) provided that no more than 20% of the interval values in the measurement are missing.

  VEE Group: Interval Estimations

  VEE Rule: INTERVAL_PROFILE_EST

  Sequence: 10

  Description: Interval Profile Estimation

  Category: Estimation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Interval Profile Estimation:

  - **Maximum Percentage Missing Intervals**: 20
  - **Estimate if Not Attached to SP**: Estimate
  - **Factor**: kWh - Profile
  - **Condition Value for Estimates Created**: System Estimate

  Insufficient Input Data Exception:

  - **Exception Type**: Insufficient Input Data
  - **Exception Severity**: Information

  Max Percentage Missing Intervals Exception:

  - **Exception Type**: Maximum Percentage of Missing Data Exceeded
  - **Exception Severity**: Information

Scalar Calculation From Interval

Scalar Calculation From Interval rules calculate a scalar value from measurements for a related interval measuring component for the same time period as the initial measurement being estimated.

- **Rule Name**: Scalar Calculation From Interval
- **Base Package VEE Rule Business Object**: D2-ScalarCalcFromInterval
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-SCACALINT
- **Apply VEE Rule Algorithm Parameters**: N/A
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
    - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.
    - **IMD Created Condition Value**: The condition value assigned to estimated scalar data created by this rule.
- **Processing Logic**:

  Scalar Calculation From Interval rules calculate a single consumption amount for a scalar initial measurement using the total consumption for the same date/time range for a related interval measuring component.

  The calculated scalar value replaces any existing value within the initial measurement (in the post-VEE list) and updates the condition to the value defined by the "IMD Created Condition Value" parameter".

  If this rule is executed for an interval measuring component, the rule does not produce an error, and does not attempt to perform any processing.
- **Example**: The following VEE rule calculates a scalar measurement value based on the total consumption for the same date/time range for a related interval measuring component.

  VEE Group: s

  VEE Rule: SCALAR_CALC_FROM_INTERVAL

  Sequence: 10

  Description: Scalar Calculation From Interval

  Category: Estimation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Scalar Calculation From Interval:

    - **IMD Created Condition Value**: System Estimate

  Insufficient Input Data Exception:

    - **Exception Type**: Insufficient Input Data
    - **Exception Severity**: Information

rules estimate scalar usage based on Average Daily Use (ADU) for the same measuring component from one year-ago or from its most recent measurements. If historical data is not available for the previous year, the rule will use the most recent reading for the initial measurement's measuring component. Rules of this type can be configured to specify to specify a required percentage of historical data for estimation, the maximum percentage of user-edited or system estimated historical allowed in estimation calculations, and high and low tolerances for estimated values relative to earlier measurements for the same measuring component.

- **Rule Name**:
- **Base Package VEE Rule Business Object**: D2-ScalarEstimation
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-SCALAREST
- **Apply VEE Rule Algorithm Parameters**:
    - Condition - bottom range value for System-Estimated
    - Condition - top range value for System-Estimated
    - Condition - bottom range value for Regular
    - Condition - top range value for Regular
    - Condition - bottom range value for No Read
    - Condition - top range value for No Read
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
    - : Defines the parameters used by this rule, including:
        - **Historical Percentage Required**: The percentage amount of history required for the validation to be executed.
        - **Historical Look First**: Specifies the first point in time to find historical data (Last Reading, Last Year)
        - **Historical Pre-Window**: The number of days prior to the start date to find historical data.
        - **Historical Post-Window**: The number of days after the stop date to find historical data.
        - **Allowed Historical System Estimated Data (%)**: The percentage of system estimated historical data allowed in the estimation calculation.
        - **Allowed Historical User Edited Data (%)**: The percentage of user edited historical data allowed in the estimation calculation.
        - **Interim Reading High Threshold Percentage**: When estimating consumption for a date/time after which there are existing measurements, this field defines the percentage by which an estimated consumption value can be greater than the consumption derived from an interpolated reading for that date/time. This value can be greater than 100%.
        - **Interim Reading Low Threshold Percentage**: When estimating consumption for a date/time after which there are existing measurements, this field defines the percentage by which an estimated consumption value can be less than the consumption derived from an interpolated reading for that date/time.
        - **Condition Value for Estimates Created**: The condition value assigned to estimated interval data values created by this rule.
    - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.
    - **Estimated Value Exceed Limit Exception**: Defines the Exception Type and Severity for exceptions created when the estimated value exceeds the limits based on existing reading.

- **Processing Logic**:

 rules use historical data for the same measuring component to derive an estimated value for a scalar initial measurement. The rule can use historical data from either the previous year or the previous reading, based on the “Historical Look First” parameter.

If the data for the selected historical period (Last Reading or Last Year) turns out to be unusable, the second historical period is evaluated. Whether historical data qualifies for use in estimation is determined through a combination of rule parameters and a set of algorithm parameters.

The “Condition - bottom range value for System Estimated” and the “Condition - top range value for System Estimated” parameters define the condition range for system estimated intervals, while the “Condition - bottom range value for Regular” and the “Condition - top range value for Regular” parameters define the condition range for regular intervals The rule does not attempt to estimate if there is already a value present in the initial measurement that has a condition that falls within the ranges defined as either "system-estimated" or "regular" based on the algorithm parameters.

Estimated values are calculated based on whether or not the measuring component measures a “peak quantity” (such as demand measured by kilowatts).

- If the measuring component references a unit of measure that does not "measure peak quantity", the rule calculates an estimated value by finding an "average daily usage" for the historical period, and applying that value to the length of the current period of the initial measurement. For example, if the initial measurement is 30 days, and the calculated average daily usage is 50, the initial measurement would be equal to 1500 (30 days X 50 per day).
- If the measuring component references a unit of measure that "measures peak quantity", the routine uses the maximum returned from the historical period as the estimated value.

In both cases, the rule calls a common routine to arrive at the estimated value that is also used by the High/Low Check.

The rule rejects consumption from a historical period as unusable for estimation if too great a portion of the period is covered by final measurements that are not high-quality. The “Allowed Historical System Estimated Data (%)” and “Allowed Historical User Edited Data (%)” parameters define the maximum percentage of system estimated or user edited historical data (respectively) that is allowed to be used in estimation calculations.

Once an estimated value is calculated, the routine backs into a reading, which involves backing out multipliers and, if the measuring component is subtractive, adding the result to the prior reading. The Calculate Scalar Consumption (D1-SC-CNSUMP) algorithm does this in a special "Back Into Reading" mode.

Assuming the measuring component does not "measure peak quantity" (in other words, assuming it measures consumption), and that it does not allow negative consumption, the rule then validates that the current estimate is in line with any subsequent measurement (if one exists), and compares it to a “reasonable” value, as follows

- If the estimate is greater than the consumption represented by the subsequent measurement, an exception is created and the estimated value is not used. To find the consumption represented by the subsequent measurement, the rule calls the Calculate Scalar Consumption algorithm using the start reading of the current initial measurement and end reading of the subsequent measurement.
- If the estimate does not exceed the consumption represented by the subsequent measurement, the rule evaluates whether the estimate exceeds a "reasonable" value by some percentage. The "reasonable" value is found by calculating an average daily usage for the consumption represented by the subsequent measurement, and multiplying this average by the number of days of the current initial measurement. The percentage difference is found by taking the difference between the "reasonable" value and the

estimate, and dividing by the reasonable value. If the resulting percentage exceeds the value specified for the "Interim High Reading Threshold" parameter, the estimate is rejected and an exception is logged.

If the estimate passes these checks, the Initial Measurement is updated with its value, and the condition is updated with the new condition specified by the "Condition Value for Estimates Created" parameter.

- **Example**: The following VEE rule estimates scalar usage based on measurements from one year-ago, providing that no more than 20% of the historical data was either user-edited to estimated.

  VEE Group: s

  VEE Rule: SCALAR_ESTIMATION

  Sequence: 10

  Description:

  Category: Estimation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  :

  - **Historical Percentage Required**: 50
  - **Historical First Look**: Last Year
  - **Historical Pre-Window**: 10
  - **Historical Post-Window**: 5
  - **Allowed Historical System Estimated Data (%)**: 20
  - **Allowed Historical User Edited Data (%)**: 20
  - **Interim Reading High Threshold Percentage**: 50
  - **Interim Reading Low Threshold Percentage**: 50
  - **Condition Value Estimates Created**: System Estimate

  Insufficient Input Data Exception:

  - **Exception Type**: Insufficient Input Data
  - **Exception Severity**: Information

  Estimated Value Exceeds Limit Exception:

  - **Exception Type**: Consumption Differs from Reference Amount
  - **Exception Severity**: Information

Scalar Profile Estimation

Scalar Profile Estimation rules estimate missing scalar readings based on an associated profile measuring component. Rules of this type can be configured to specify to specify a required percentage of historical data for estimation, the maximum percentage of system estimated historical allowed in estimation calculations, and high and low tolerances for estimated values relative to earlier measurements for the same measuring component.

- **Rule Name**: Scalar Profile Estimation
- **Base Package VEE Rule Business Object**: D2-ScalarProfileEstimation
- **Apply VEE Rule Algorithm Type / Algorithm**: D2-SCAPROEST

- **Apply VEE Rule Algorithm Parameters**:
  - Condition bottom range value for System-Estimated
  - Condition top range value for System-Estimated
  - Condition code bottom range value for Regular
  - Condition code top range value for Regular
  - No Read Bottom Range Condition
  - No Read Top Range Condition
- **Validation Algorithm Type / Algorithm**: N/A
- **Rule Parameters**:
  - **Scalar Profile Estimation**: Defines the parameters used when estimating missing scalar data from a profile, including:
    - **Historical Percentage Required**: The percentage amount of history required for the validation to be executed.
    - **Allowed Historical System Estimated Data (%)**: The percentage of system estimated historical data allowed in the estimation calculation.
    - **Estimation Profile Factor**: Defines the profile factor referencing the measuring component to use for generating estimates if no historical data for the same measuring component can be found. If it's not populated, profile-based estimation will not be performed when no historical data is found to be usable from the current measuring component.
    - **Interim Reading High Threshold Percentage**: When estimating consumption for a date/time after which there are existing measurements, this element defines the percentage by which an estimated consumption value can be greater than the consumption derived from an interpolated reading for that date/time. This value can be greater than 100%.
    - **Interim Reading Low Threshold Percentage**: When estimating consumption for a date/time after which there are existing measurements, this element defines the percentage by which an estimated consumption value can be less than the consumption derived from an interpolated reading for that date/time.
    - **Condition Value for Estimates Created**: The condition value assigned to estimated interval data values created by this rule.
  - **Insufficient Input Data Exception**: Defines the Exception Type and Severity for exceptions created when there is insufficient data to execute the VEE rule.
  - **Estimated Value Exceed Limit Exception**: Defines the Exception Type and Severity for exceptions created when the estimated value exceeds the limits based on existing reading.
- **Processing Logic**:

  Scalar Profile Estimation rules calculate a scalar estimate by looking at final measurements for a profile measuring component covering the same date range as the current initial measurement. The profile measuring component to be used as a source of measurement data is defined in the "Estimation Profile Factor" parameter. This rule is meant primarily for a configuration in which the profile measuring components are interval, although the profile could be scalar as well.

  The "Condition bottom range value for System Estimated" and the "Condition top range value for System Estimated" algorithm parameters define the condition range for system estimated intervals, while the "Condition code bottom range value for Regular" and the "Condition code top range value for Regular" algorithm parameters define the condition range for regular intervals The rule does not attempt to estimate if there is already a value present in the initial measurement that has a condition that falls within the ranges defined as either "system-estimated" or "regular" based on the algorithm parameters.

These algorithm parameters are also used in the evaluation of the quality of the profile measuring component's data. The "system estimated" condition range algorithm parameter values are used in conjunction with the "Allowed Historical System Estimated Data (%)" parameter to reject the profile data if too high a percentage of the data has a condition inside the "system estimated" range.

If the measuring component references a unit of measure that "measures peak quantity", the maximum value returned from the profile data is used as the estimated value. Otherwise, an "average daily usage" value is calculated using the profile data, and that value is applied to the length of the current period of the initial measurement. For example, if the initial measurement is 30 days, and the calculated average daily usage is 50, the initial measurement would be equal to 1500 (30 days X 50 per day).

Note that the logic in this algorithm calls the same common routine used in the High/Low Check and rules.

- **Example**: The following VEE rule estimates scalar measurement values based on a specified profile measuring component, providing that no more than 20% of the historical data was estimated.

  VEE Group: s

  VEE Rule: SCALAR_PROFILE_EST

  Sequence: 10

  Description: Scalar Profile Estimation

  Category: Estimation Rules.

  Start Date: 05-01-2010

  End Date: N/A

  Scalar Profile Estimation:

  - **Historical Percentage Required**: 50
  - **Allowed Historical System Estimated Data (%)**: 20
  - **Estimation Profile Factor**: kWh - Profile
  - **Interim Reading High Threshold Percentage**: 50
  - **Interim Reading Low Threshold Percentage**: 50
  - **Condition Value for Estimates Created**: System Estimate

  Insufficient Input Data Exception:

  - **Exception Type**: Insufficient Input Data
  - **Exception Severity**: Information

  Estimated Value Exceeds Limit Exception:

  - **Exception Type**: Consumption Differs from Reference Amount
  - **Exception Severity**: Information

VEE Eligibility Criteria In Detail

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

- A description of the D1-VEEELIGCR maintenance object
- Lists of the base package VEE eligibility criteria business objects, including "lite" business objects
- Details concerning VEE eligibility criteria-specific configuration options
- A sample VEE eligibility criteria business object (D1-VEEEligibilityCriteria)

Maintenance Object - D1-VEEELIGCR

VEE eligibility criteria business objects use the D1-VEEELIGCR maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-VEEELIGCR
Description	VEE Eligibility Criteria
Service Name	D1-VEEELIGCR (VEE Eligibility Criteria)
Tables	• D1_VEE_ELIG_CRIT (VEE Eligibility Criteria) - Primary • D1_VEE_ELIG_CRIT_CHAR (VEE Eligibility Criteria Characteristics) - Child • D1_VEE_ELIG_CRIT_L (VEE Eligibility Criteria Language) - Child

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

Meter Data Framework Base Package VEE Eligibility Criteria Business Objects

The meter data framework base package includes the following VEE eligibility criteria business objects:

Business Object Name	Description
D1-VEEEligibilityCriteria	VEE Eligibility Criteria Instances of this business object represent individual eligibility criteria defined in the system.

The meter data framework base package includes the following additional VEE eligibility criteria business objects:

Business Object Name	Description
D1-VEEEligCritBundlingAddBO	Bundling Add BO for VEE Eligibility Criteria
D1-VEEEligCritPhysicalBO	Physical BO for VEE Eligibility Criteria

Configuration Options

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

System Events

VEE eligibility criteria business objects can make use of the following system events:

- **Apply VEE Rule Eligibility Criteria**: This system event defines the algorithm to use to apply eligibility criteria to a VEE rule.

Example VEE Eligibility Criteria - D1-VEEEligibilityCriteria

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

Option/Field	Description
Business Object	D1-VEEEligibilityCriteria
Description	VEE Eligibility Criteria
Maintenance Object	D1-VEEELIGCR (VEE Eligibility Criteria)
Application Service	D1-VEEELIGCR(VEE Eligibility Criteria MO)
Instance Control	Allow New Instances
Options	• Portal Navigation Option: d1veerleTabMenu (VEE Rule) • Maintenance UI Map: D1-VEEEligibilityCritMaint (VEE Eligibility Criteria - Maintenance)
Algorithms	• Apply VEE Rule Eligibility Criteria: D1-ECF-APECT (Evaluate Criteria Field - Apply Eligibility Criteria)

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

VEE Exceptions In Detail

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

- A description of the D1-VEEEXCP maintenance object
- Lists of the base package VEE exception business objects, including "lite" business objects
- A sample VEE exception business object (D1-VEEException)

Maintenance Object - D1-VEEEXCP

VEE exceptions business objects use the D1-VEEEXCP maintenance object. The table below outlines some of the details of this maintenance object.

Option/Field	Description
Maintenance Object	D1-VEEEXCP
Description	VEE Exception
Service Name	D1-VEEEXCP (VEE Exception Maintenance)
Tables	• D1_VEE_EXCP (VEE Exception) - Primary • D1_VEE_EXCP_CHAR (VEE Exception Characteristics) - Child • D1_VEE_EXCP_PARM (VEE Exception Parameter) - Child

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

Meter Data Framework Base Package VEE Exception Business Objects

The meter data framework base package includes the following VEE exception business objects:

Business Object Name	Description
D1-VEEException	VEE Exception

Example VEE Exception - D1-VEEException

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

Option/Field	Description
Business Object	D1-VEEException
Description	VEE Exception
Maintenance Object	D1-D1-VEEEXCP (VEE Exception)
Application Service	D1-D1-VEEEXCP (VEE Exception MO)
Instance Control	Allow New Instances

Option/Field	Description
Options	• Display UI Map: D1-VEEExceptionDisplay (VEE Exception - Display) • Display Map Service Script: D1-VEEExcptn (Display VEE Exception Message Details)
Algorithms	• Information: D1-VEXCPINFO (VEE Exception Information)

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

Configuring VEE Groups, Rules, Eligibility Criteria, and Exceptions

This section provides high-level overviews of the steps involved in configuring custom VEE groups and rules. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

This section also provides information related to creating instances of the “generic utility” VEE rules described earlier in this chapter.

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

Configuring Custom VEE Groups

Configuring custom VEE groups involves the following steps:

1. Design the VEE group business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom VEE group-related configuration objects required for your business objects.
3. Create your VEE group business objects, referencing the configuration objects created above as appropriate.

Configuring Custom VEE Rules

Configuring custom VEE rules involves the following steps:

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

   System Events: Create algorithms for the following system events:

   - Apply VEE Rule

   Options: Create data as appropriate for the following options used when creating VEE rules:

   - Exception Types
   - Generic Utility VEE Rules: Define the following options used with “generic utility” VEE rules
     - Referred VEE Group: VEE groups to be referenced by these rules.
     - Exception Handler: To Do Type, To Do Role, and Exception Type used by these rules.
     - VEE Group Matrix (Factor): Factor, Characteristic Type and Values, Characteristic Source Algorithm, VEE groups
3. Create your VEE rule business objects, referencing the configuration objects created above as appropriate.

   Note: VEE rule business objects should reference D1-GenericVEERule as their Parent Business Object.

Configuring Custom VEE Eligibility Criteria

Configuring custom VEE eligibility criteria involves the following steps:

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

   System Events: Create algorithms for the following system events:

   - Apply VEE Rule Eligibility Criteria
3. Create your VEE eligibility criteria business objects, referencing the configuration objects created above as appropriate.

Configuring Custom VEE Exceptions

Configuring custom VEE exceptions involves the following steps:

1. Design the VEE exception business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom VEE exception-related configuration objects required for your business objects.
3. Create your VEE exception business objects, referencing the configuration objects created above as appropriate.

Creating Generic Utility VEE Rules

This section outlines the steps involved in creating instances of the “generic utility” VEE rules. Refer to the Oracle Utilities Meter Data Framework online help and user’s guide for general procedures used in creating VEE rules.

Creating Execute VEE Group VEE Rules

Use the following procedure to create Execute VEE Group VEE rules:

1. Create the VEE group to which the rule will belong.
2. Create the VEE group that the rule will reference
3. Create the VEE rule referencing the group created in the previous step

Creating VEE Group Matrix (Factor) VEE Rules

Use the following procedure to create VEE Group Matrix (Factor) VEE rules:

1. Create the Characteristic Type and Values to be used by the factor that will be referenced by the rule.
2. Create the Characteristic Source Algorithm to be used by the factor that will be referenced by the rule.
3. Create the VEE Groups to be associated to the characteristic values.
4. Create the Factor that will be referenced by the rule.
5. Create the Factor Values for the factor, each referencing an effective-dated characteristic value/VEE group pairings.
6. Create the rule, referencing the factor

Creating Exception Handler VEE Rules

Use the following procedure to create Exception Handler VEE rules:

1. Create the To Do Type to be used by the rule.
2. Create the To Do Role to be used by the rule.
3. Create the Exception Type to be used by the rule.
4. Create the rule, including:
    - To Do Type, To Do Role, and Exception Type created in the previous steps.
    - Criteria comparison for the rule.

Chapter 8

Device Communication and Device Events

This chapter provides descriptions of entities related to device communications, including activities, communications, and completion events. This chapter also describes entities related to device events. This chapter includes:

- **Understanding Device Communication**
- **Understanding Device Events**
- **Inbound Communications in Detail**
- **Outbound Communications in Detail**
- **Completion Events in Detail**
- **Device Events in Detail**
- **Configuring Device Communication and Device Event Objects**

Note: The command, communication, and completion event functionality described in this chapter is available only with Oracle Utilities Smart Grid Gateway.

Understanding Device Communication

This section provides an overview of entities related to device communications, including activities, communications, and completion events and how they are used in the meter data framework and related products, including Oracle Utilities Meter Data Management and Oracle Utilities Smart Grid Gateway.

Activities

Activities are records of a communication related to a device, measuring component, 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. Activities are used extensively 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 Gatewaty 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 **Undestanding 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**
- **Outbond 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

Outbond 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 **Undestanding 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 Comunication 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 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)

Step	Process	Landis+Gyr Adapter Objects
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 Comunication BO: Initiate Connect Disconnect (D3-InitiateConnectDisconnect)
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 Comunication 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 Comunication BO: Initiate Connect Disconnect (D3-InitiateConnectDisconnect)

Understanding Device Events

This section provides an overview of device events and how they are used in the meter data framework 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 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 meter data framework 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 meter data framework 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.

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 meter data framework

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

Step	Process	Meter Data Framework 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 SGG format.	
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)
4.	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.	Processing Method: D1-HowToProcessDeviceInfo (How to Process Device Related Information) Device Event Mapping BO: D1-DeviceEventMappingLookup

Step	Process	Meter Data Framework Objects
5.	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)
6.	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 “Sent to Subscribers” 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)

Time Zone Translation

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.

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-RemoteConnect)

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.

Meter Data Framework Base Package Activity Business Objects

The meter data framework base package includes the following 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-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-EnableService	Enable Service
D1-GenericConnect	Generic Connect Device
D1-GenericDisconnect	Generic Disconnect Device
D1-GenericReadDevice	Generic Read Device
D1-MeterReadDownloadActivity	Meter Read Download Activity Instances of this business object represent individual meter read downloads in the system.
D1-MultiDeviceStatusCheck	Multi-Device Status Check Instances of this business object represent individual multi-device status check (ping) commands initiated in the system
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.

Business Object Name	Description
D1-OnDemandReadScalar	On-Demand Read Scalar Instances of this business object represent individual scalar on-demand read commands initiated in the system.
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.
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.
D1-SPActivityOrchestration	SP Activity Orchestation
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 meter data framework base package includes the following “lite” activity business objects:

Business Object Name	Description
D1-ActivityBIBOToRead	Activity BI BO To Read
D1-ActivityLite	Activity LITE

Configuration Options

This section outlines specific types of BO Options used by activity business objects.

Business Object Options

Activity business objects 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 options:

Lookup FIeld Value	Description
D1AM	Obtain AMI Device Identifier
D1DA	Activity Notification
D1DC	Device Commission
D1DD	Device Decommission
D1DM	Device Event Mapping
D1DS	Device Status Check
D1EP	Event Processing Default Configuration
D1FR	Response - Fail
D1IM	Initial Measurement Creation
D1IN	On-Demand Read (Interval)
D1LC	Load Check
D1RC	Remote Connect
D1RD	Remote Disconnect
D1RM	Retrieve Meter
D1RR	Response - Received
D1SC	On-Demand Read (Scalar)
D1SD	Send Device Event
D1SR	Response - Success
D1UM	UOM Mapping

Example Activity - D1-RemoteConnect

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

Option	Description
Business Object	D1-RemoteConnect
Description	Remote Connect
Maintenance Object	D1-ACTIVITY (Activity)
Application Service	D1-REMOTECONNECTBOAS (Remote Connect BO)

Option	Description
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)
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)

Option	Description
Lifecycle	• Pending (Initial) • Validation (Interim) • Validation Error (Interim) • Waiting for Effective Date (Interim) • Connection Ready (Interim) • Communication in Progress (Interim) • Communication Error (Interim) • Retry (Interim) • Execute Completion Events (Interim) • Completion Events Error (Interim) • Waiting for Measurement (Interim) • Wait Expired (Interim) • Completed (Final) • Discarded (Final)

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

Meter Data Management Base Package Activity Business Objects

The Oracle Utilities Meter Data Management base package includes the following activity business objects. These are used to create aggregator measuring components. See **Automatic Creation of Aggregator Measuring Components** on page 13-7 for more information.

Business Object Name	Description
D2-ActivityAggDimScanner	Aggregator Creator - Postal/Service Type
D2-MsrmtQuantityAggScanner	Measurement Quantity Scanner

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.

Meter Data Framework Base Package Inbound Communication Business Objects

The meter data framework 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.

Meter Data Framework Base Package Outbound Communication Business Objects

The meter data framework 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.

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.

Meter Data Framework Base Package Completion Event Business Objects

The meter data framework 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)
D1-ConnectDevice	Connect Device Completion Event Instances of this business object represent individual connect device completion events in the system.

Business Object Name	Description
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 - Retreive 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.

Meter Data Framework Base Package Device Event Business Objects

The meter data framework 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 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-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.

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 busines 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	Allow New Instances
Options	None (options are defined for child business objects)
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) • Sent to Subscribers (Final)

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

Configuring Device Communication and Device Event Objects

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

   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 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.

Chapter 9

Usage Subscriptions

This chapter provides descriptions of usage subscriptions and usage subscription types, including:

- **Understanding Usage Subscriptions**
- **Usage Subscriptions In Detail**
- **Usage Subscription Types In Detail**
- **Configuring Usage Subscriptions and Usage Subscription Types**

Understanding Usage Subscriptions

This section describes usage subscriptions and their role in the usage calculation process.

Usage Subscriptions

Oracle Utilities Meter Data Management can calculate and send bill determinants to other systems, such as a customer information system (CIS), an external billing system, or some other application. Before bill determinants can be calculated, you must first create a usage subscription. A usage subscription can be thought of as an ongoing request to send one or more service points' usage to one or more external systems, and defines the service point’s bill determinants are calculated.

Bill determinants (usage) are derived from the final measurements of the measuring components installed at the usage subscription's service points during the calculation period. A service point is linked to measuring components through a install event linked to the measuring components’ device configuration.

An Aside: No Account Object Exists

Oracle Utilities Meter Data Management (and related meter data products) is not considered the system of record for accounts or usage subscriptions. The customer information system (or some other system) is considered the system of record for this type of information. In order to minimize the amount of data that must be synchronized between systems, account-oriented attributes used by the meter data products are held on usage subscriptions. For example, if an account’s ID and its customer class are relevant to usage calculations, each usage subscription must reference both elements. This is an important distinction to keep in mind when creating custom usage subscriptions for your implementation.

Multiple Service Points and Multiple Measuring Components

At any instance in time:

- A usage subscriptions may be linked to multiple service points.
- A service point may be linked to a single device configuration
- A device configuration may have multiple measuring components

The calculation period for bill determinant calculations can span many days and over this period:

- The service points linked to the usage subscription can change (service points can be added and removed)
- The device configurations installed at the service point can change (due to device reconfigurations and meter exchanges)

This means that values for each bill determinants can be calculated using multiple service points and measuring components.

Contacts

Contacts are individuals or business entities with which a company has contact. A contact exists for every individual or business related to a usage subscription. A single usage subscription can have many contacts, and a single contact may be referenced on many different usage subscriptions. Contacts have a 1-to-1 correlation with a "person" in a customer information system (CIS) and the CIS is considered the system of record for contact information.

Service Points and Contacts

As noted in the description of service points, service points can reference contacts. While this is optional, all usage subscriptions must reference at least one contact.

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

Service Providers

As noted in the service point chapter, service providers can be associated with a market and/or the service points in a market. This is optional and typically only set up in deregulated markets.

Usage subscriptions, on the other hand, must reference a service provider. The service provider is used as the identity of the subscribing system. In other words, you must set up a service provider for any system that subscribes to bill determinants.

Usage Subscriptions In Detail

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

- A description of the D1-US maintenance object
- Lists of the base package usage subscription business objects, including “lite” business objects
- Details concerning usage subscription-specific configuration options
- A sample usage subscription business object (D2-UsageSubscription)

Maintenance Object - D1-US

Usage subscription business objects use the D1-US maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-US
Description	Usage Subscription
Service Name	D1-US (Usage Subscription Maintenance)
Tables	• D1_US (Usage Subscription) - Primary • D1_US_CHAR (Usage Subscription Characteristics) - Child • D1_US_CONTACT(Usage Subscription - Contact) - Child • D1_US_FACTOR_OVRD (Usage Subscription Factor Override) - Child • D1_US_IDENTIFIER (Usage Subscription Identifier) - Child • D1_US_LOG (Usage Subscription Log) - Child • D1_US_LOG_PARM (Usage Subscription Log Parameters) - Child • D1_US_SP (Usage Subscription - Service Point) - Child • D1_US_SP_CHAR (Usage Subscription - SP Characteristics) - Child • D1_US_USG_GRP (Usage Subscription - Usage Group) - Child

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

Meter Data Management Base Package Usage Subscription Business Objects

The Oracle Utilities Meter Data Management base package includes the following usage subscription business objects:

Business Object Name	Description
D2-UsageSubscription	Usage Subscription Instances of this business object represent individual usage subscriptions defined in the system.

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

Business Object Name	Description
D2-USMainContact	Usage Subscription’s Main Contact LITE (used to retrieve the usage subscription's main contact)
D2-USMainContactLITE	Usage Subscription Main Contact LITE (used to retrieve the usage subscription's main contact)
D2-UsageSubscription-Ext-Char	Usage Subscription LITE
D2-UsageSubscriptionParentLITE	Usage Subscription Parent LITE
D2-UsgTranProUsgSub	Usage Subscription Transaction Processing (describes the structure and rules applicable to Usage Subscription Transaction)
D2-USSPLite	Usage Subscription Service Point Lite

The Oracle Utilities Meter Data Management base package includes the following additional usage subscription business objects:

Business Object Name	Description
D2-SynchronizationAddUS	US Synchronization Add (used when adding a new usage subscription as a result of a data synchronization request)

Example Usage Subscription - D2-UsageSubscription

The table below lists the details of the D2-UsageSubscription usage subscription business object.

Option/Field	Description
Business Object	D2-UsageSubscription
Description	Usage Subscription
Maintenance Object	D1-US (Usage Subscription)
Application Service	D2-USBOAS (Usage Subscription BO)
Instance Control	Allow New Instances

Option/Field	Description
Options	• Synchronization Add BO: D2-SynchronizationAddUS (US Synchronization Add) • Display UI Map: D2-USDisplay (Usage Subscription - Display) • Portal Navigation Option: d1usmTabMenu (Usage Subscription) • Display Map Service Script: D2-USRetDtls (Usage Subscription Retrieve Details for Display) • Maintenance UI Map: D2-USMaint (Usage Subscription - Maintenance)
Algorithms	• Information: D2-USINFO (Usage Subscription Information) • Pre-Processing: D1-DEFTIMZON (Default Time Zone value based on Installation Option) • Validation: D2-USSPPRVAL (Validate that US can not be linked to Non-Parent Service Points) • Validation: D2-USSPDTVAL (Validate Start and Stop Dates of Usage Subscription’s Service Points) • Validation: D2-USUGDTVAL (Validate Effectivity Period of Usage Subscription’s Usage Groups) • Validation: D2-USFOVAL (Validate the Start and Stop Dates of US’ Factor Overrides) • Validation: D1-VALTIMZON (Validates BO Time Zone value against Installation Option
Lifecycle	• Active (Initial) • Inactive (Final)

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

Usage Subscription Types In Detail

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

- A description of the D1-USTYPE maintenance object
- Lists of the base package usage subscription type business objects, including “lite” business objects
- Details concerning usage subscription type-specific configuration options
- A sample usage subscription type business object (D2-UsageSubscriptionType)

Maintenance Object - D1-USTYPE

Usage subscription type business objects use the D1-USTYPE maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-USTYPE
Description	Usage Subscription Type
Service Name	D1-USTYPE (Usage Subscription Type Maintenance)
Tables	• D1_US_TYPE (Usage Subscription Type) - Primary • D1_US_TYPE_CHAR (Usage Subscription Type Characteristics) - Child • D1_US_TYPE_FB_USG_GRP (Usage Subscription Type - Fallback Usg Grp) - Child • D1_US_TYPE_L (Usage Subscription Type - Language) - Child • D1_US_TYPE_VAL_SPR (Usage Subscription Type - Valid Service Provider) - Child • D1_US_TYPE_SP_TYPE (Usage Subscription Type - Valid Valid SP Type) - Child • D1_US_TYPE_VAL_USG_GRP (Usage Subscription Type - Valid Usage Group) - Child

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

Meter Data Management Base Package Usage Subscription Type Business Objects

The Oracle Utilities Meter Data Management base package includes the following usage subscription type business objects:

Business Object Name	Description
D2-UsageSubscriptionType	Usage Subscription Type Instances of this business object represent individual usage subscription types defined in the system.

The Oracle Utilities Meter Data Management base package includes the following “lite” usage subscription type business objects:

Business Object Name	Description
D1-UsageSubscriptionTypeLite	Usage Subscription Type - LITE
D2-UsageSubscriptionTypeLITE	Usage Subscription Type LITE
D2-USTypeParentLITE	Usage Subscription Type Parent LITE
D2-UsgTranProSubTyp	Usage Subscription Type Transaction LITE (describes the structure and rules applicable to Usage Subscription Type Transaction)

The Oracle Utilities Meter Data Management base package includes the following additional usage subscription type business objects:

Business Object Name	Description
D1-UsageSubscrTypeBundlingAddBO	Bundling Add BO for Usage Subscription Type
D1-UsageSubscrTypeBO	Physical BO for Usage Subscription Type

Configuration Options

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

Other Options

Usage subscription types define many attributes of the usage subscriptions of that type. These options are specified when creating usage subscriptions based on a usage subscription business object, and include the following:

Valid Service Point Types

These define the service point types considered valid for usage subscriptions of this type.

Valid Service Providers

These define the service providers considered valid for usage subscriptions of this type.

Usage Groups (Valid and Fallback)

Usage Groups define the usage rules to be applied to initial measurement data for usage subscriptions of this type.

- **Valid Usage Groups**: These define the usage groups considered valid for measuring components of this type.

- **Fallback Usage Groups**: These define the usage groups that can be used with all usage subscriptions of this type in situations where the usage groups defined for the usage subscriptions are not in effect. at the time usage is to be calculated Fallback usage groups have effective dates which define the point in time after which they are considered in effect.

Example Usage Subscription Type - D2-UsageSubscriptionType

The table below lists the details of the D2-UsageSubscription usage subscription type business object.

Option/Field	Description
Business Object	D2-UsageSubscriptionType
Description	Usage Subscription Type
Maintenance Object	D2-USTYPE (Usage Subscription Type)
Application Service	D2-USTYPE (Usage Subscription Type BO)
Instance Control	Allow New Instances
Options	• Display UI Map: D2-USTypeDisplay (Usage Subscription Type - Display) • Portal Navigation Option: d1ustypeTabMenu (Usage Subscription Type) • Maintenance UI Map: D2-USTypeMaint (Usage Subscription Type - Maintenance)

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

Configuring Usage Subscriptions and Usage Subscription Types

This section provides high-level overviews of the steps involved in configuring custom usage subscriptions and usage subscription types. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

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

Configuring Custom Usage Subscriptions

Configuring custom usage subscriptions involves the following steps:

1. Design the usage subscription business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom usage subscription-related configuration objects required for your business objects.
3. Create your usage subscription business objects, referencing the configuration objects created above as appropriate.

Configuring Custom Usage Subscription Types

Configuring custom usage subscription types involves the following steps:

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

   Options: Create data as appropriate for the following options used when creating usage subscription types:

   - Valid Service Point Types
   - Valid Service Providers
   - Usage Groups (Valid and Fallback)
3. Create your usage subscription type business objects, referencing the configuration objects created above as appropriate.
4. Set up admin records that define the usage subscription types you will use in your implementation.

Chapter 10

Usage Groups and Usage Rules

This chapter provides descriptions of usage groups and rules, including:

- **Understanding Usage Groups and Rules**
- **Usage Groups In Detail**
- **Usage Rules In Detail**
- **Usage Eligibility Criteria In Detail**
- **Configuring Usage Rules, Groups, and Eligibility Criteria**

Understanding Usage Groups and Rules

This section describes the usage calculation process used Oracle Utilities meter data framework and Oracle Utilities Meter Data Management, including descriptions of usage groups and rules.

The Usage Calculation Process

Oracle Utilities Meter Data Management can calculate and publish usage calculated from measurement data to service providers on an ongoing basis. In addition, external systems can request usage whenever needed. Usage calculations derive a usage transaction's usage quantities using the measurements linked to a usage subscription's service points.

The usage calculation engine and process is very similar to the VEE engine in that it is driven by configurable rules. These rules calculate a usage transaction's usage (also known as bill determinants). Usage calculation rules can also be configured to validate the usage that was calculated by earlier rules. If problems are found, exceptions are created and the usage transaction is not finalized.

Most requests for usage result in the creation of a usage transaction, but it is possible for an external application to invoke the usage calculation engine real-time. In other words, usage can be retrieved for a usage subscription real-time without creating a usage transaction. This technique is only recommended for online requests, not as part of batch processes.

See **Chapter 12**: **Usage Transactions** for more information about the usage calculation process.

Usage Rules and Usage Groups

The specific usage calculation processing performed on (final) measurement data is defined in individual usage rules, each performing a specific set of calculations. The base package includes rules that calculate common bill determinants including:

- Scalar reads
- Time-of-use consumption (by applying a time-of-use map to an interval channel)
- Interval curves (either real or derived)
- Virtually anything else that can be calculated from the information in the system

The base package contains several usage rules you can use in your implementation, and you can also create your own custom usage rules based on your requirements.

Usage Groups

Usage groups are collections of usage rules that are applied to measurement data. During the usage calculation process, the system executes the usage rules defined in each usage group. The rules within a usage group are defined in a specific sequence, allowing control over the order in which the rules are executed.

Fallback and Valid Usage Groups

Usage groups can be associated to a specific usage subscription, or to a usage subscription type (or both). Usage groups associated to a usage subscription type are considered “fallback” usage groups. In addition, a usage subscription’s usage subscription type also defines the usage groups that can be defined for individual usage subscriptions, and that are considered “valid” to override the fallback usage groups. Only the valid usage groups on the usage subscription type can be referenced on an individual usage subscription. An individual usage subscription can have override usage groups. If the usage subscription doesn't have an override usage group for a bill determinant's calculation period, the fallback group defined on the usage subscription type is used.

Effective Dates

Unlike VEE rules, usage rules do not have individual effective dates. Usage groups associated to a usage subscription have effective and expiration dates, which define the date range during which they can applied to usage calculations for the usage subscription. A usage subscription can have many usage groups where each has a different effective period (and multiple usage groups can be effective during a usage transaction's period). In other words, the entire usage group is effective-dated rather than the individual rules.

> **Note**: In MDM 2.0.0, the system uses the group effective at the start of the calculation period. In MDM 2.0.1, the system calculates separate sets of bill determinants for each group in effect during the usage period

Eligibility Criteria

Each usage rule may optionally have eligibility criteria that controls if the rule is applied. This feature can greatly reduce the number of usage groups you need to create, because it allows a single usage group to have conditional usage rules based on eligibility criteria (rather than requiring a distinct usage group for every combination of v rules).

For example, you might use eligibility criteria that specifies that a rule is only applied if the customer has solar power (or some other unique characteristic).

Referred Usage Group Rules - Reusing Groups Of Rules

Just like VEE rules, a usage rule can reference a different usage group, so commonly used rules can be encapsulated in reusable usage groups. A usage rule that executes the rules in a referenced usage group is called an Execute Usage Group rule. Rules of this type can have effective eligibility criteria, just like all usage rules.

Execute Usage Group rules can be “nested.” That is, a group executed by a Execute Usage Group rule can, in turn, execute the rules in another group, and so on.

Using Factors For Variables

A situation common in many implementations involves converting one unit of measure (UOM) to another. However, the conversion factor used in conversions of this can differ based on many different types of criteria, such as the location of the service point or other characteristics. This sort of calculation can be implemented as a usage rule that accumulates consumption for one UOM and converts the consumption to a different UOM by applying a factor to it.

Factors used for this purpose have a Factor Class of “Number,” and use some unique rules:

- Number factors reference a characteristic type (with pre-defined values).
- Number factors reference an algorithm that retrieves or derives the value of the characteristic type at runtime.
- Factor values for a Number factor are effective-dated pairings of a characteristic value and a corresponding value. Because these pairings are effective-dated, the value returned from the factor can change over time for each characteristic value

At run time, the rule retrieves / derives the characteristic value for the factor's characteristic type and then finds the value associated with the respective characteristic value.

Factors can be related to any real or dynamic attribute, so rules of this type are very flexible. For example:

- **Real Attribute**: you could create a rule that retrieves a specific value based on the location of a service point.
- **Dynamic Attribute**: you could create a rule that retrieves a percentage value based on the amount the customer conserved as compared to the same period in the prior year, returning one value if the amount conserved is between 5% and 10%, another value if the amount conserved is between 10% and 20%, and yet a third value if the amount conserved is greater

than 20%. The amount conserved is dynamically calculated at execution time and is compared to the characteristic values defined for the factor, and returns the appropriate value. In this example, if the amount conserved was anything less than 5%, no percentage value would be returned.

Usage Groups In Detail

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

- A description of the D1-USGGRP maintenance object
- Lists of the base package usage group business objects, including "lite" business objects
- A sample usage group business object (D1-UsageGroup)

Maintenance Object - D1-USGGRP

Usage group business objects use the D1-USGGRP maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-USGGRP
Description	Usage Group
Service Name	D1-USAGEGRP (Usage Group Maintenance)
Tables	• D1_USG_GRP (Usage Group) - Primary • D1_USG_GRP_CHAR (Usage Group Characteristics) - Child • D1_USG_GRP_L (Usage Group Language) - Child • D1_USG_GRP_VAL_DC_TYPE (Usage Group - Valid DC Types) - Child

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

Meter Data Management Base Package Usage Group Business Objects

The Oracle Utilities Meter Data Management base package includes the following usage group usage business objects:

Business Object Name	Description
D1-UsageGroup	Usage Group Instances of this business object represent individual usage groups defined in the system.

The Oracle Utilities Meter Data Management base package includes the following additional usage group business objects:

Business Object Name	Description
D1-UsageGroupBundlingAddBO	Bundling Add BO for Usage Group
D1-UsageGroupPhysicalBO	Physical BO for Usage Group

Example Usage Group - D1-UsageGroup

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

Option/Field	Description
Business Object	D1-UsageGroup
Description	Usage Group
Maintenance Object	D1-USGGRP (Usage Group)
Application Service	D1-USAGEGRP (Usage Group MO)
Instance Control	Allow New Instances
Options	• Display UI Map: D1-UsageGroupDisplay (Usage Group - Display) • Portal Navigation Option: d1usggrpTabMenu (Usage Group) • Maintenance UI Map: D1-UsageGroupMaint (Usage Group - Maintenance)

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

Usage Rules In Detail

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

- A description of the D1-USGRULE maintenance object
- Lists of the base package usage rule business objects, including “lite” business objects
- Details concerning usage rule-specific configuration options
- A sample usage rule business object (D2-ApplyMathInt)
- A list of base package usage rules, including the algorithm / algorithm type and a brief description of each

Maintenance Object - D1-USGRULE

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

Option/Field	Description
Maintenance Object	D1-USGRULE
Description	Usage Rule
Service Name	D1-USAGERULE (Usage Rule Maintenance)
Tables	• D1_USG_RULE (usage Rule) - Primary • D1_USG_RULE_CHAR (usage Rule Characteristics) - Child • D1_USG_RULE_L (usage Rule Language) - Child

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

Meter Data Framework Base Package Usage Rule Business Objects

The meter data framework base package includes the following usage rule business objects:

Business Object Name	Description
D1-GenericUsageRule	Generic Usage Rule (generic usage rule business object used as a parent business object for all other usage rule business objects)
D1-UsageRuleReferredUsageGroup	Execute Usage Group Instances of this business object represent individual Execute Usage Group usage rules defined in the system.

The meter data framework base package includes the following “lite” usage rule business objects:

Business Object Name	Description
D1-UsageRuleExeReSequenceLite	Usage Rule Execution Resequencing LITE

Business Object Name	Description
D1-UsageRuleParentLITE	Usage Rule Parent LITE

The meter data framework base package includes the following additional usage rule business objects:

Business Object Name	Description
D1-UsageRuleBundlingAddBO	Bundling Add BO for VEE Rule
D1-UsageRulePhysicalBO	Physical BO for VEE Rule

Meter Data Management Base Package Usage Rule Business Objects

The Oracle Utilities Meter Data Management base package includes the following usage rule business objects:

Business Object Name	Description
D2-ApplyMathInt	Apply Math (Interval Data)
D2-GetIntervalData	Get Interval Data
D2-GetScalar	Get Scalar Details
D2-GetTOUUsage	Get TOU Mapped Usage
D2-Math	Math
D2-ValAgainstTol	Validate Against Tolerance

Configuration Options

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

System Events

Usage rule business objects can make use of the following system events:

- **Apply Usage Rule**: This system event defines the algorithm to use when executing the usage rule.
- **Validation**: This system event defines the algorithm to use to validate the measurement data used by the usage rule.

Other Options

Usage rules use various parameters and properties. These options are specified when creating usage rules based on a usage rule business object, and include the following:

Generic Utility VEE Rules

The meter data framework includes one "generic utility" base package usage rule type that can be used when configuring usage groups and rules. This section outlines the configuration options you need to configure before you can create rules of these types.

Execute Usage Group: Execute Usage Group rules reference a usage group. You must create the usage group to reference before can create rules of this type.

Example Usage Rule - D2-ApplyMathInt

The table below lists the details of the D2-ApplyMathInt device business object.

Option/Field	Description
Business Object	D2-ApplyMathInt
Description	Apply Math (Interval Data)
Maintenance Object	D1-USGRULE (Usage Rule)
Parent Business Object	D1-GenericUsageRule (Generic Usage Rule)
Lifecycle Business Object	Generic Usage Rule
Application Service	D1-USAGERULE (Usage Rule MO)
Instance Control	Allow New Instances
Options	• Display UI Map: D2-ApplyMathIntDisplay (Usage Rule-Apply Math (Interval Data) Display) • Portal Navigation Option: d1usgrleTabMenu (Usage Rule) • Maintenance UI Map: D2-ApplyMathIntMaint (Usage Rule-Apply Math (Int) - Maintenance)
Algorithms	• Apply Usage Rule: D2-APPMATHIN Apply Math (interval data) • Validation: D2-VALAPPMTH (Apply Math on Interval Data - Validation)

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

Base Package Usage Rules Summary

The following table lists the back package usage rules. Each of these usage rules is provided as a business object and corresponding algorithm/algorithm type.

Usage Rule	Business Object	Algorithm / Algorithm Type	Description
Generic Usage Rule	D1-GenericUsageRule	N/A	Parent usage rule for all other usage rules, contains basic usage rule schema elements
Apply Math (Interval Data)	D2-ApplyMathInt	D2-APPMATHIN	A usage rule used to summarize interval measurements and apply a mathematical formula against the results to derive a usage quantity.
Get Interval Data	D2-GetIntervalData	D2-GETINTDAT	A usage rule used to get interval data for a measuring component and date range.
Get Scalar Details	D2-GetScalar	D2-GETSCALAR	A usage rule used to assemble scalar readings and measurements (consumption).
Get TOU Mapped Usage	D2-GetTOUUsage	D2-GETTOUUSG	A usage rule used to summarize interval measurements into TOU buckets based on a TOU map (where the TOU map is created based on the schedule defined within a TOU map template).
Math	D2-Math	D2-MATH	A usage rule used to derive a curve of interval values given a formula, apply TOU mapping to a derived curve, and perform mathematical operations on Usage Transaction Service Quantity entries.
Validate Against Tolerance	D2-ValAgainstTol	D2-VALUSGTOL	A usage rule used to validate calculated usage against a specified tolerance.

Use the Business Object portal and Algorithm portal (or Application Viewer) to view additional details about these usage rules.

Usage Rule Descriptions

This section provides detailed descriptions of each of the base package usage rules provided with Oracle Utilities Meter Data Management. The descriptions of the usage rules below contain the following information

- **Rule Name**: The name of the rule
- **Rule Description**: A brief description of the rule
- **Base Package Usage Rule Business Object**: The base package business object used by the rule
- **Apply Usage Rule Algorithm Type / Algorithm**: The base package algorithm type and algorithm specified in the "Apply Usage Rule" system event for the rule
- **Apply Usage Rule Algorithm Parameters**: The "soft" parameters (if any) used by the "Apply Usage Rule" algorithm used by the rule
- **Validation Algorithm Type / Algorithm**: The base package algorithm type and algorithm specified in the "Validation" system event for the rule, if applicable
- **Rule Parameters**: Parameters used to define the rule (if applicable). Does not include common parameters used by all rules (see **Common Parameters** on page 10-11).
- **Processing Logic**: A description of the processing performed by the rule.
- **Example**: One or more example configurations of the rule. Note that all examples use default values for all "Apply Usage Rule Algorithm Parameters".

Common Parameters

All usage rules use the following common parameters:

- **Usage Group**: The usage group to which the rule belongs
- **Usage Rule**: The user-defined name given to the usage rule
- **Sequence**: The sequence within its usage group that the rule will be executed
- **Description**: A short description of the rule. This is used as the Information String for the rule in the user interface.
- **Detailed Description**: A detailed description of the rule
- **Usage Rule Category**: The rule's category. Base package options include "Usage Calculation" and "Validation".

Usage Rules

This section provides descriptions for the base package usage rules used when calculating usage for usage subscriptions. Usage rules are standard and custom rules that perform calculation on measurement data to generate bill determinants and other values used by external systems, such as billing systems, customer information systems, etc. The results of usage rule calculations are stored in usage transactions as Service Quantities. Base package usage rules include:

- **Apply Math (Interval Data)**
- **Get Interval Data**
- **Get Scalar Details**
- **Get TOU Mapped Usage**
- **Math**
- **Validate Against Tolerance**

Apply Math (Interval Data)

Apply Math (Interval Data) rules summarize interval measurements and apply a mathematical formula against the results to calculate usage quantities.

- **Rule Name**: Apply Math (Interval Data)
- **Base Package Usage Rule Business Object**: D2-ApplyMathInt
- **Apply Usage Rule Algorithm Type / Algorithm**: D2-APPMATHIN
- **Algorithm Parameters**: N/A
- **Validation Algorithm Type / Algorithm**: D2-VALAPPMTH

  This algorithm performs the following validations:

  - A UOM and/or TOU and/or SQI must be supplied in the “Result” section.
  - If the Calculation Type is “Single Value”, a Single Value Variable must be supplied.
  - If the Calculation Type is “Math Function”, a Math Function and one or more Member Variables must be supplied.
  - If the Calculation Type is “Mathematical Expression”, a Mathematical Expression must be supplied.
  - For each entry in the Variables list
    - If the Variable Type is “Channel Accumulation”, a UOM and/or TOU and/or SQI and a Set Function must be supplied.
    - If the Variable Type is “Factor”, a Factor must be supplied
- **Rule Parameters**:
  - **Result**: Specifies the resulting entries the rule will insert into the usage transaction service quantity list. Options include:

    UOM: the Unit of Measure to be used when inserting service quantity entries

    TOU: the Time of Use to be used when inserting service quantity entries

    SQI: the Service Quantity Identifier to be used when inserting service quantity entries
  - **Calculation Details**: This section defines details related to the type of calculation performed, including arguments and expressions used in calculations.

    Calculation Type: Defines the type of operation to perform using variables defined in the Variables section. Options include:

    - **Single Value**: The rules returns the value of a single variable defined in the Single Value Variable field.
    - **Math Function**: The rule returns the Sum, Maximum, or Minimum value based on variables defined in the Member Variables field.
    - **Mathematical Expression**: The rule returns the result of a mathematical expression defined in the Mathematical Expression field.

    Single Value Variable: The Variable Name of the variable used by Single Value calculations. Example: V1

    Member Variables: The Variable Names, separated by a space, of the variables used by Math Function calculations. Example: V1 V2 V3

    Mathematical Expression: The mathematical expression used by Mathematical Expression calculations.

    Example 1: (V1 + V2) / V3

    Example 2: SQRT(V1^2 + V2^2)

- **Variables**: This section defines the variables used in calculations defined in the Calculation Details section.

  Variable Name: The name of the variable.

  Variable Type: The type of variable. Options include:

  - **Channel Accumulation**: Used to reference one or more interval measuring components (linked to the subscription's service points) that share common traits (for example, all measuring components with a given UOM). The UOM and/or TOU and/or SQI and a Set Function are required for this type of variable.
  - **Factor**: Used when a factor is used in the calculation.
  - **Usage Period Hours**: Used when total number of hours in the usage period is used in the calculation.

  UOM: Specifies the UOM of the measurement data when the Variable Type is Channel Accumulation.

  TOU: Specifies the TOU of the measurement data when the Variable Type is Channel Accumulation.

  SQI: Specifies the SQI of the measurement data when the Variable Type is Channel Accumulation.

  Set Function: The function to apply to the measurement data when the Variable Type is Channel Accumulation. Options include:

  - **Sum**: Returns the sum of the UOM/TOU/SQI specified. This is used when calculating the total usage for an interval measurement.
  - **Max**: Returns of the interval maximum value for the UOM specified.
  - **Min**: Returns the minimum interval value for the UOM specified.

  Target Unit of Measure: The target UOM of the calculation when the Variable Type is Channel Accumulation.

  Factor: The factor to retrieve when the Variable Type is Factor.

- **Processing Logic**:

  Apply Math (Interval Data) rules perform calculations on interval data and store the results in the usage transaction's service quantities. Calculations are based on calculation type and variables defined in the rule.

  Calculation Types can be one of the following:

  - Single Value: For Single Value calculations, the Single Variable Name is required, which identifies the variable to process.
  - Math Function: For Math Function cal cu a lt ions, a Math Function is required and one or more Member Variables are required. These define the function to apply to the Member Variables specified.
  - Mathematical Expression: For Mathematical Expression cal cu a lt ions, the Mathematical Expression is required, which identifies the mathematical expression and variables to use.

  Variables can be one of the following types:

  - Channel Accumulation: For Channel Accumulation variables, a UOM/TOU/SQI is required, which is used to identify the measuring components (linked to the usage subscription's service points) to process. A Target Unit Of Measure is optional, and is used UOM conversions, such as converting KWH to KW.

    - Factor: For Factor variables, a Factor is required that identifies the factor to use when retrieving the variable value.
    - Usage Period Hours: Usage Period Hours variables are used when the number of hours in the usage period is a variable to be used in the calculation.

  The rule applies the calculation type to the specified variables to calculate usage quantities.

- **Example**: The following usage rule calculates usage based on an interval measurement by calculating the total of the interval values for measuring components that measure KWH.

  Usage Group: Interval Usage

  Usage Rule: TOTAL_KWH

  Sequence: 10

  Description: Total KWH

  Category: Usage Calculation

  Result

  - **UOM**: Kilowatt-Hours
  - **TOU**:
  - **SQI**:

  Calculation Details:

  - **Calculation Type**: Single Value
  - **Single Value Variable**: V1

  Variables:

  - **V1**:
    - **Variable Type**: Channel Accumulation
    - **UOM**: Kilowatt-Hours
    - **Set Function**: Sum

- **Example**: The following usage rule calculates usage based on the higher value between measuring components that measure KWH and KVARH.

  Usage Group: Interval Usage

  Usage Rule: CALC_MAX

  Sequence: 10

  Description: Calculate Maximum Value for KWH or KVARH

  Category: Usage Calculation

  Result

  - **UOM**: Kilowatt-Hours
  - **TOU**:
  - **SQI**:

  Calculation Details:

  - **Calculation Type**: Math Function
  - **Math Function**: Max
  - **Member Variables**: V1 V2

Variables:

- **V1**:
    - **Variable Type**: Channel Accumulation
    - **UOM**: Kilowatt-Hours
    - **Set Function**: Sum
- **V2**:
    - **Variable Type**: Channel Accumulation
    - **UOM**: Kilovolt-Ampere Reactive Hours
    - **Set Function**: Sum

- **Example**: The following usage rule calculates usage based on an interval measurement by multiplying the total of the interval values in the measurement by a "power factor" stored as a factor.

  Usage Group: Interval Usage

  Usage Rule: POWER_FACTOR

  Sequence: 10

  Description: Apply Power Factor to Interval Data Total

  Category: Usage Calculation

  Result

  - **UOM**: Kilowatt-Hours
  - **TOU**:
  - **SQI**: Power Factor Applied

  Calculation Details:

  - **Calculation Type**: Mathematical Expression
  - **Mathematical Expression**: V1*V2

  Variables:

  - **V1**:
    - **Variable Type**: Channel Accumulation
    - **UOM**: Kilowatt-Hours
    - **Set Function**: Sum
  - **V2**:
    - **Variable Type**: Factor
    - **Factor**: Power Factors

Get Interval Data

Get Interval Data rules calculate interval data quantities from interval measurements for a specified UOM, TOU, or SQI.

- **Rule Name**: Get Interval Data
- **Base Package Usage Rule Business Object**: D2-GetIntervalData
- **Apply Usage Rule Algorithm Type / Algorithm**: D2-GETINTDAT
- **Algorithm Parameters**: N/A
- **Validation Algorithm Type / Algorithm**: **Rule**: D2-VALINTDAT

  This algorithm performs the following validations:

  - A UOM and/or TOU and/or SQI must be supplied.
- **Parameters**:
  - **Interval Data Details**: Specifies the resulting interval data entries the rule will insert into the usage transaction service quantity list. Options include:
    - **UOM**: The unit of measure to be retrieved from the interval measurement. Should be used only with measuring components that measure one or more UOMs.
    - **TOU**: The time of use to be retrieved from the interval measurement. Should be used only with measuring components that measure one or more TOUs.
    - **SQI**: The service quantity identifier to be retrieved from the interval measurement. Should be used only with measuring components that measure one or more SQIs.
    - **Calculation Function**: How usage is calculated from the interval data (Max or Sum).
- **Processing Logic**:

  Get Interval Data rules get interval quantities from interval measuring components installed in the service points linked to the usage subscription for the specified 'interval' usage period. Only measuring components that match the UOM/TOU/SQI defined in the usage rule are processed.

  Measurements within the period are stored in the usage transaction's service quantities' interval data list. The service quantity entry's quantity is calculated based on the Calculate Function (Max or Sum) defined in the usage rule. This is done for every entry in the usage period list.
- **Example**: The following usage rule calculates the sum of the interval values in interval measurements that record kilowatt hours.

  Usage Group: Interval Usage

  Usage Rule: GET_INTERVAL_KWH DATA

  Sequence: 10

  Description: Get Interval Data - KWH

  Category: Usage Calculation

  Interval Data Details:

  - **UOM**: Kilowatt-Hours
  - **TOU**:
  - **SQI**:
  - **Calculation Function**: Sum

Get Scalar Details

Get Scalar Details rules assemble scalar readings and measurements.

- **Rule Name**: Get Scalar Details
- **Base Package Usage Rule Business Object**: D2-GetScalar
- **Apply Usage Rule Algorithm Type / Algorithm**: D2-GETSCALAR
- **Algorithm Parameters**:
    - Estimate Bottom Range Condition
    - Estimate Top Range Condition
    - Measurement Cycle Schedule Thru Date Option (1 - Usage Period End Date, 2 - Retry Until Date)
    - Characteristic Type To Identify "Summary Account" Usage Subscriptions
    - Characteristic Value To Identify "Summary Account" Usage Subscriptions
- **Validation Algorithm Type / Algorithm**: **Rule**: D2-VALSCALAR

  This algorithm performs the following validations:
    - A UOM and/or TOU and/or SQI must be supplied.
- **Rule Parameters**:
    - **Scalar Details**: Defines specific UOMs, TOUs, or SQIs to be retrieved by the usage rule, and if the results should be added to the service quantity (SQ) list for the usage period. If not specified, the rule processes all scalar measuring components for the usage transaction's service point.
- **Processing Logic**:

  Get Scalar Details rules get usage from scalar measuring components installed in the service points linked to the usage subscription for the specified 'scalar' usage period.

  By default all scalar measuring components are processed, but if specific UOMs/TOUs/SQIs are defined in the usage rule, then only applicable measuring components are processed. Measurements within the usage period are retrieved. The usage transaction request may indicate whether or not 'Estimate' measurements are allowed. The "Estimate Bottom Range Condition" and "Estimate Top Range Condition" algorithm parameters are used to define the range of measurement condition values that correspond to 'Estimate' measurements.

  The measurement details are stored in the usage transaction's scalar details. The usage is also stored in the usage transaction's service quantities unless otherwise specified in the usage rule (using Build Service Quantity indicator).

  The "Measurement Cycle Schedule Thru Date Option (1 - Usage Period End Date, 2 - Retry Until Date)", "Characteristic Type To Identify "Summary Account" Usage Subscriptions", and "Characteristic Value To Identify "Summary Account" Usage Subscriptions" paramters are used when integrating Oracle Utilities Meter Data Management to Oracle Utilities Customer Care and Billing. The "Measurement Cycle Schedule Thru Date Option (1 - Usage Period End Date, 2 - Retry Until Date)" parameter is used to define the thru date for measurement cycles. Available values are:

  1. Usage Period End Date
  2. Retry Until Date (the bill cycle window end date)

  The "Characteristic Type To Identify "Summary Account" Usage Subscriptions", and "Characteristic Value To Identify "Summary Account" Usage Subscriptions" parameters are used to define the characteristic type and value (respectively) used as indicators in custom algorithms used to trigger estimation prior to the bill cycle window end date.

- **Example**: The following usage rule retrieves kilowatt hours from scalar measurements and adds the results to the Service Quantity list in the usage transaction.

  Usage Group: Scalar Usage

  Usage Rule: GET_SCALAR_KWH

  Sequence: 10

  Description: Get Scalar KWH

  Category: Usage Calculation

  Scalar Details:

  - **UOM**: Kilowatt-Hours
  - **TOU**:
  - **SQI**:
  - **Build Service Quantity**: Yes (checked)

Get TOU Mapped Usage

Get TOU Mapped Usage rules calculate usage from interval measurements based on a TOU map. Service quantities are created for each TOU period defined for the TOU map used by the rule. For example, usage calculated from a TOU map with "On-Peak" and "Off-Peak" TOU periods would result in both "On-Peak" and "Off-Peak" service quantities.

- **Rule Name**: Get TOU Mapped Usage
- **Base Package Usage Rule Business Object**: D2-GetTOUUsage
- **Apply Usage Rule Algorithm Type / Algorithm**: D2-GETTOUUSG
- **Algorithm Parameters**:
- **Validation Algorithm Type / Algorithm**: **Rule**: D2-VALTOUUSG

  This algorithm performs the following validations:

  - UOM and/or SQI must be supplied
- **Rule Parameters**:
  - **TOU Mapping Details**: Specifies details of how interval measurements are mapped into TOU quantities, including:
    - **Result SQI**: The resulting SQI for entries inserted into the service quantities list by the rule
    - **Unit of Measure**: The UOM used to filter the measuring components from which interval measurements are used by the rule.
    - **Service Quantity Identifier**: The SQI used to filter the measuring components from which interval measurements are used by the rule.
    - **Time of Use Calculate Function**: The function used to calculate TOU data (Max or Sum)
    - **TOU Map**: The ID of the TOU map used.
- **Processing Logic**:

  Get TOU Mapped Usage rules get time of use quantities from interval measuring components installed in the service points linked to the usage subscription for the specified 'interval' usage period. Only measuring components that match the UOM/SQI defined in the usage rule instance are processed.

  Measurements within the period are mapped to time of use quantities based on the TOU map defined in the “TOU Map” parameter. If dynamic options are specified in the referenced TOU map and if there are dynamic option events in effect within the usage period, the TOU map associated with the dynamic option is used for the entire dynamic option event period. This is done for every usage period requested.

  The calculated time of use quantities are stored in the usage transaction's service quantities.
- **Example**: The following usage rule calculates TOU values from interval measurements that record kilowatt hours based on a specified TOU map.

  Usage Group: Interval Usage

  Usage Rule: GET_TOU_MAPPED_KWH

  Sequence: 10

  Description: Get TOU Mapped KWH Usage

  Category: Usage Calculation

  TOU Mapping Details:

  - **Result SQI**:

- **Unit of Measure**: Kilowatt-Hours
- **Service Quantity Identifier**:
- **Time of Use Calculate Function**: Sum
- **TOU Map**: Summer / Winter, 15 minute interval

Math

Math rules derive interval data measurements based on a formula, and apply TOU mappings and/or other operations to the derived data to calculate usage quantities. Math rules can also perform operations on existing Usage Transaction Service Quantity entries.

- **Rule Name**: Math
- **Base Package Usage Rule Business Object**: D2-Math
- **Apply Usage Rule Algorithm Type / Algorithm**: D2-MATH
- **Algorithm Parameters**:
- **Validation Algorithm Type / Algorithm**: **Rule**: D2-VALMATH

  This algorithm validates the Math entries.
- **Rule Parameters**:
  - **Vector 1 (Vector 2, Vector 3, ... Vector 5)**: Defines the vectors to be used in the calculation. When used in formulas, interval values for this vector are designated as IV1, IV2, IV3, IV4, or IV5.

    Type: the source of values for this vector:
    - **Channels Linked To Usage Subscription**: retrieves values from channels linked to the usage subscription (via its related service point, and their related devices and measuring components), based on specified UOM, TOU, and SQI
    - **Profile Factor**: retrieves values from the profile measuring component defined on the profile factor
    - **Specific Measuring Component**: retrieves values from a specified measuring component
    - **Usage Transaction Service Quantity**: retrieves values from the usage transaction service quantity entries, identified by UOM, TOU, and SQI.

    Unit of Measure: the UOM of the values to be retrieved. This is applicable if type is Channels Linked to Usage Subscription or Usage Transaction Service Quantity. If type is either Profile Factor or Specific Measuring Component, this is only applicable if Use Primary Measurement is set to No.

    Time of Use: the TOU of the values to be retrieved. This is applicable if type is Channels Linked to Usage Subscription or Usage Transaction Service Quantity. If type is either Profile Factor or Specific Measuring Component, this is only applicable if Use Primary Measurement is set to No.

    Service Quantity Identifier: the SQI for the values to be retrieved. This is applicable if type is Channels Linked to Usage Subscription or Usage Transaction Service Quantity. If type is either Profile Factor or Specific Measuring Component, this is only applicable if Use Primary Measurement is set to No.

    Target Unit of Measure: if specified, the retrieved values are further converted. In order to perform UOM conversion, the UOM and Target UOM must have the same Base UOM. For example, KWH and MWH. This is applicable only if type is Channels Linked To Usage Subscription.

Profile Factor: the profile factor used to retrieve the profile measuring component from which the values are retrieved. This is applicable only if type is Profile Factor.

Profile: if specified, this is the profile (based on the profile factor above) used to retrieve the profile measuring component. This is applicable only if type is Profile Factor.

Use Primary Measurement: indicates that the primary measurement values measured by the referenced measuring component are to be used in calculations. This is applicable only if type is either Profile Factor or Specific Measuring Component

Measuring Component ID: the measuring component from which the values are retrieved. This is applicable only if type is either Profile Factor or Specific Measuring Component.

- **Scalar Variables**: Defines scalar variables to be used in the calculation. When used in formulas, scalar variables are designated as Vn, where n is the number of the variable (based on the sequence in which they are defined in the list).

  Type: the type of variable. Can be one of the following:

  - **Factor**: the value for a specified factor that is in effect for the usage period.
  - **Set Function**: the result of applying a function to a set of interval measurement values (defined as a vector).
  - **Usage Transaction Service Quantity**: service quantities from the current usage transaction, based on a specified UOM, TOU, or SQI. This type of variable allows this rule to make use of values calculated by other rules.

  Set Function: the function used to calculate a scalar value from a set of interval measurement values (defined as a vector). This is applicable only if Type is Set Function). Can be one of the following:

  - **Average**: calculates the average of the vector's interval measurement values.
  - **Count**: returns the number of interval measurements.
  - **Max**: returns the maximum value from the vector's interval measurement values.
  - **Min**: returns the minimum value from the vector's interval measurement values.
  - **Total**: calculates the total of the vector's interval measurement values.

  Interval Set: the vector to be used for this variable (applicable only if Type is Set Function). Can be one of the following:

  - **FV (Final Vector Interval Value)**: the vector containing the results of the formula defined in the Vector Processing section.
  - **IV1 (Vector 1 Interval Value)**: the vector defined as Vector 1.
  - **IV2 (Vector 2 Interval Value)**: the vector defined as Vector 2.
  - **IV3 (Vector 3 Interval Value)**: the vector defined as Vector 3.
  - **IV4 (Vector 4 Interval Value)**: the vector defined as Vector 4.
  - **IV5 (Vector 5 Interval Value)**: the vector defined as Vector 5.

  Factor: the factor used to retrieve the variable value (applicable only if Type is Factor).

  Unit of Measure: the UOM for service quantities to retrieve from the usage transaction's service quantity list (applicable only if Type is Usage Transaction Service Quantity).

  Time of Use: the TOU for service quantities to retrieve from the usage transaction's service quantity list (applicable only if Type is Usage Transaction Service Quantity).

Service Quantity Identifier: the SQI for service quantities to retrieve from the usage transaction's service quantity list (applicable only if Type is Usage Transaction Service Quantity).

- **Vector Processing**: Defines how to calculate the interval values for the curve/vector to be derived. This is not applicable when performing purely scalar calculations.

  Common Interval Size: the common interval size (designated as hours:minutes:seconds) to which interval measurement values (from defined vectors) should be scaled before performing calculations. This is required if multiple vectors are defined.

  Vector Formula Source: the type of formula to be used to derive the interval values. Can be one of the following:

  - **Simple Vector Formula**: indicates that a simple formula will be used.
  - **Conditional Vector Formula**: indicates that a conditional formula will be used. This allows comparison between one or more pairs of operands to determine the specific formula to execute.

  Simple Vector Formula: the simple formula used to derive the interval values. Can reference a vector (designated as IVn, where n is the number of the vector) or an expression referencing one or more vectors or a scalar variable (designated as Vn).

  Conditional Vector Formula: the conditional formula used to derive the interval measurement values. A conditional formula can utilize one or more conditions. Each condition comprises the following:

  - **Operand 1**: the first operand in the condition. Can reference a vector (designated as IVn, where n is the number of the vector) or an expression referencing one or more vectors or a scalar variable (designated as Vn).
  - **Criteria Operator**: the operator used to compare Operand 1 with Operand 2.
  - **Operand 2**: the second operand in the condition. Can reference a vector (designated as IVn, where n is the number of the vector) or an expression referencing one or more vectors or a scalar variable (designated as Vn).
  - **True Action**: indicates how to proceed if the comparison between the operands is true. Can be one of the following:
    - **Apply True Formula**: indicates that the True Formula be executed.
    - **Check Next Condition**: indicates that the next condition should be checked.
  - **True Formula**: the formula to apply if True Action is set to Apply True Formula. Can reference a vector (designated as IVn, where n is the number of the vector) or an expression referencing one or more vectors or a scalar variable (designated as Vn).
  - **False Action**: indicates how to proceed if the comparison between the operands is false. Can be one of the following:
    - **Apply False Formula**: indicates that the False Formula be executed.
    - **Check Next Condition**: indicates that the next condition should be checked.
  - **False Formula**: the formula to apply if the False Action is set to Apply False Formula. Can reference a vector (designated as IVn, where n is the number of the vector) or an expression referencing one or more vectors or a scalar variable (designated as Vn).

- **Result**: This usage rule can insert one or more entries into the usage transaction service quantity list.

  Unit of Measure: the UOM to be used when inserting service quantity entries

Service Quantity Identifier: the SQI to be used when inserting service quantity entries

Insert SQ Entry: indicates whether or not to insert an entry into the usage transaction service quantity list. This should always be set to Yes if Apply TOU Map To Derived Vector is set to No.

SQ Entry Quantity Source: indicates the method to use to calculate for the service quantity (applicable only if Insert SQ Entry is Yes). Can be one of the following:

- **Set Function Against Derived Vector**: applies a function to the derived interval measurement values. The function to be applied is specified in the Set Function Against Derived Vector field.
- **Scalar Formula Result**: applies a user-defined formula. The formula is specified in the Scalar Formula field.

Set Function Against Derived Vector: the function to apply to the derived interval measurement values (applicable only if SQ Entry Quantity Source is set to Set Function Against Derived Vector). Can be one of the following:

- **Average**: calculates the average of the derived interval measurement values.
- **Count**: returns the number of derived interval measurements.
- **Max**: returns the maximum value from the derived interval measurement values.
- **Min**: returns the minimum value from the derived interval measurement values.
- **Total**: calculates the total of the derived interval measurement values.

Scalar Formula: the formula to apply (applicable only if SQ Entry Quantity Source is set to Scalar Formula). Variables used in this formula must be defined in the Scalar Variables section. When referenced in formulas, scalar variables are designated as Vn (where n is the number of the variable).

Save Derived Vector: indicates whether or not to store the derived interval measurement values in the usage transaction service quantity entry to be inserted (applicable only if Insert SQ Entry is set to Yes).

Apply TOU Map To Derived Vector: indicates if a TOU map should be applied to the derived interval measurement values. If TOU periods and values are returned as a result of TOU mapping, then service quantity entries are inserted regardless on how Insert SQ Entry is set.

TOU Map: the TOU Map to apply to the derived interval measurement values (applicable only if Apply TOU Map To Derived Vector is set to Yes).

Time of Use Calculate Function: the function to apply to the derived interval measurement values when calculating for the time of use values (applicable only of Apply TOU Map To Derived Vector is set to Yes). Can be one of the following:

- **Max**: returns the maximum value from the derived interval measurement values for each TOU period.
- **Sum**: returns the sum of the derived interval measurement values for each TOU period.

- **Processing Logic**:

  Math rules derive interval data measurements based on a formula, and apply TOU mappings and/or other operations to the derived data to calculate usage quantities. Examples include the following:

  - Derive an interval data curve (vector) given a formula. For example, derive a power factor curve given a formula using kWh and kvarh curves.

- Apply TOU mapping to a derived interval data curve. For example, after deriving the power factor curve, perform TOU mapping on the result.
- Perform mathematical operations on Usage Transaction Service Quantity entries. For example, get total kWh consumption by adding "kWh On Peak", "kWh Off Peak", and "kWh Shoulder Peak" where "kWh On Peak", "kWh Off Peak", and "kWh Shoulder Peak" were calculated by a previous usage rule.

Each interval data curve is defined as a vector parameter (the rule can define up to 5 vectors). Mathematical operations defined by the "Vector Processing" parameters can be performed between vectors (e.g. IV1 * IV2) and between vectors and scalar variables (e.g. IV1 * V1). When performing mathematical operations on more than one vector, the "Common Interval Size" parameter specifies the interval size that each vector is to be scaled to prior to calculations. For example, in the above example (IV1 * IV2), if vector IV1 contains 15 minute intervals, and vector IV2 contains 1 hour intervals, the 15 minute vector is scaled to 1 hour intervals prior to multiplying the two vectors. The values used for service quantities are derived from the resulting derived vector.

- **Example**: The following usage rule calculates KVA (kilovolt amperes) from KW (kilowatts) and KVAR (kilovolt amperes reactive).

Usage Group: Interval Usage

Usage Rule: CALC_KVA

Sequence: 10

Description: Calculate KVA from KW and KVAR

Category: Usage Calculation

Vector 1: (IV1)

- **Type**: Channels Linked To Usage Subscription
- **Unit of Measure**: Kilowatts
- **Time of Use**:
- **Service Quantity Identifier**:
- **Target Unit of Measure**: Kilowatts

Vector 2: (IV2)

- **Type**: Channels Linked To Usage Subscription
- **Unit of Measure**: Kilovolt-Amperes Reactive
- **Time of Use**:
- **Service Quantity Identifier**:
- **Target Unit of Measure**: Kilovolt-Amperes Reactive

Scalar Variables: N/A

Vector Processing:

- **Common Interval Size**: 01:00:00
- **Vector Formula Source**: Simple Vector Formula
- **Simple Vector Formula**: (IV1=IV2)

Result

- **Unit of Measure**: Kilovolt-Amperes
- **Service Quantity Identifier**:

- **Insert SQ Entry**: Yes
- **SQ Entry Quantity Source**: Set Function Against Derived Vector
- **Set Function Against Derived Vector**: Max
- **Save Derived Vector**: Yes
- **Apply TOU Map to Derived Vector**: No

- **Example**: The following usage rule calculates net usage for customers with solar panels by subtracting generated usage from consumed usage, and multiplying the resulting interval values by a "power factor" stored as a factor.

  Usage Group: Interval Usage

  Usage Rule: CALC_NET_USAGE_PF

  Sequence: 10

  Description: Calculate Net Usage with Power Factor Applied

  Category: Usage Calculation

  Vector 1: (IV1)

  - **Type**: Channels Linked To Usage Subscription
  - **Unit of Measure**: Kilowatt-Hours
  - **Time of Use**:
  - **Service Quantity Identifier**: Consumed
  - **Target Unit of Measure**: Kilowatt-Hours

  Vector 2: (IV2)

  - **Type**: Channels Linked To Usage Subscription
  - **Unit of Measure**: Kilowatt-Hours
  - **Time of Use**:
  - **Service Quantity Identifier**: Generated
  - **Target Unit of Measure**: Kilowatt-Hours

  Scalar Variables:

  - **V1**:
    - **Sequence**: 1
    - **Factor**: Power Factors

  Vector Processing:

    - **Common Interval Size**: 01:00:00
    - **Vector Formula Source**: Simple Vector Formula
    - **Simple Vector Formula**: (IV1=IV2)*V1

  Result

  - **Unit of Measure**: Kilowatt-Hours
  - **Service Quantity Identifier**: Net Consumption
  - **Insert SQ Entry**: Yes
  - **SQ Entry Quantity Source**: Set Function Against Derived Vector
  - **Set Function Against Derived Vector**: Total

- **Save Derived Vector**: Yes
- **Apply TOU Map to Derived Vector**: Yes
- **TOU Map**: Year round schedule, 15 minute interval
- **Time of Use Calculate Function**: Sum

Validate Against Tolerance

Validate Against Tolerance rules validate calculated usage service quantities against a specified tolerance. These rules are typically used later in the usage calculation process to validate results of previously executed usage rules. For example, after calculating kilowatt hour usage for a particular service point, a rule of this type could be used to validate that the calculated usage does not exceed a certain threshold.

- **Rule Name**: Validate Against Tolerance
- **Base Package Usage Rule Business Object**: D2-ValAgainstTol
- **Apply Usage Rule Algorithm Type / Algorithm**: D2-VALUSGTOL
- **Algorithm Parameters**:
- **Validation Algorithm Type / Algorithm**: **Rule**: D2-VALTOL

  This algorithm performs the following validations:

  - A UOM and/or TOU and/or SQI must be supplied.
  - Either a Tolerance Value or Tolerance Factor must be supplied.
- **Rule Parameters**:
  - **Validation Details**: Defines details of how the usage is validated, including:
    - **UOM**: The UOM whose quantity is validated by the rule.
    - **TOU**: The TOU whose quantity is validated by the rule.
    - **SQI**: The SQI whose quantity is validated by the rule.
    - **Set Function**: The function (Sum, Max, or Min) used to calculate the usage value to be validated.
    - **Tolerance Value**: A user-defined value that serves as the tolerance value in the validation.
    - **Tolerance Factor**: The factor used to define the tolerance value in the validation.
    - **Comparison Operator**: The mathematical operator used to compare the usage value to the tolerance value.
    - **Exception Severity**: The severity of exceptions triggered if the usage value fails the validation.
- **Processing Logic**:

  Validate Against Tolerance rules validate the calculated usage against a tolerance value. When configuring the usage group, this usage rule must be placed after all the usage calculation usage rules.

  This rule aggregates all usage transaction service quantities that match the UOM/TOU/SQI defined in the usage rule. Depending on the function defined in the "Set Function" parameter, the aggregated value can either be the highest quantity, the lowest quantity or the sum of all quantities.

  The tolerance value may either come from the value specified in the "Tolerance Value" parameter or a factor value for the factor defined in the "Tolerance Factor" parameter.

The “Comparison Operator” parameter determines how the two values (the aggregated value and the tolerance) are compared. If comparison results to True, an entry is inserted into the Exceptions List, based in the “Exception Severity” parameter.

- **Example**: The following usage rule validates that the sum of calculated usage values for kilowatt hours is less than or equal to a specified tolerance.

  Usage Group: Interval Usage

  Usage Rule: VALIDATE_AGAINST_TOLERANCE

  Sequence: 10

  Description: Validate Against Tolerance

  Category: Usage Calculation

  Validation Details:

  - **UOM**: Kilowatt-Hours
  - **TOU**:
  - **SQI**:
  - **Set Function**: Sum
  - **Tolerance Value**: 1000
  - **Comparison Operator**: <=
  - **Exception Severity**: Information

Usage Eligibility Criteria In Detail

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

- A description of the D1-USGRLELIG maintenance object
- Lists of the base package usage rule eligibility criteria business objects, including "lite" business objects
- Details concerning usage eligibility criteria-specific configuration options
- A sample usage rule eligibility criteria business object (D1-UsgRuleEligibilityCriteria)

Maintenance Object - D1-USGRLELIG

usage eligibility criteria business objects use the D1-USGRLELIG maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-USGRLELIG
Description	Usage Rule Eligibility Criteria
Service Name	D1-USGRLELIG (Usage Eligibility Criteria Maintenance)
Tables	• D1_USG_RULE_ELIG_CRIT (Usage Rule Eligibility Criteria) - Primary • D1_USG_RULE_ELIG_CRIT_CHAR (Usage Rule EC Characteristics) - Child • D1_USG_RULE_ELIG_CRIT_L (Usage Rule Eligibility Criteria Language) - Child

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

Meter Data Framework Base Package Usage Rule Eligibility Criteria Business Objects

The meter data framework base package includes the following usage rule eligibility criteria business objects:

Business Object Name	Description
D1-UsgRuleEligibilityCriteria	Usage Rule Eligibility Criteria Instances of this business object represent individual usage rule eligibility criteria defined in the system.

The meter data framework base package includes the following additional usage rule eligibility criteria business objects:

Business Object Name	Description
D1-UsageRuleEligCritBundlingAddBO	Bundling Add BO for VEE Eligibility Criteria

Business Object Name	Description
D1-UsageRuleEligCritPhysicalBO	Physical BO for VEE Eligibility Criteria

Configuration Options

This section outlines specific configuration options, such as business object options, system events, and other options used by usage rule eligibility criteria business objects.

System Events

Usage rule eligibility criteria business objects can make use of the following system events:

- **Apply Usage Rule Eligibility Criteria**: This system event defines the algorithm to use to apply eligibility criteria to a usage rule.

Example Usage Rule Eligibility Criteria - D1-UsgRuleEligibilityCriteria

The table below lists the details of the D1-UsgRuleEligibilityCriteria usage rule eligibility criteria business object.

Option/Field	Description
Business Object	D1-UsgRuleEligibilityCriteria
Description	Usage Rule Eligibility Criteria
Maintenance Object	D1-USGRLELIG (Usage Rule Eligibility Criteria)
Application Service	D1-USGRLELIG (Usage Rule Eligibility Criteria MO)
Instance Control	Allow New Instances
Options	• Portal Navigation Option: d1usgrleTabMenu (Usage Rule) • Maintenance UI Map: D1-UsageEligibilityCritMaint (Usage Rule Eligibility Criteria Maintenance)
Algorithms	• Apply Usage Rule Eligibility Criteria: D1-ECF-APUSG (Evaluate Criteria Field - Apply Eligibility Criteria)

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

Configuring Usage Rules, Groups, and Eligibility Criteria

This section provides high-level overviews of the steps involved in configuring custom usage groups and rules. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

This section also provides information related to creating instances of the Referred Usage Group "generic utility" usage rule described earlier in this chapter.

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

Configuring Custom Usage Groups

Configuring custom usage groups involves the following steps:

1. Design the usage group business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom usage group-related configuration objects required for your business objects.
3. Create your usage group business objects, referencing the configuration objects created above as appropriate.

Configuring Custom Usage Rules

Configuring custom usage rules involves the following steps:

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

   System Events: Create algorithms for the following system events:

   - Apply Usage Rule

   Options: Create data as appropriate for the following options used when creating usage rules:

   - Generic Utility Usage Rules: Define the following options used with "generic utility" usage rules
     - Referred Usage Group: Usage groups to be referenced by these rules.
3. Create your usage rule business objects, referencing the configuration objects created above as appropriate.

   Note: Usage rule business objects should reference D1-GenericUsageRule as their Parent Business Object.

Configuring Custom Usage Rule Eligibility Criteria

Configuring custom usage eligibility criteria involves the following steps:

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

   System Events: Create algorithms for the following system events:

- Apply Usage Rule Eligibility Criteria

3. Create your usage eligibility criteria business objects, referencing the configuration objects created above as appropriate.

Creating Execute Usage Group Usage Rules

Use the following procedure to create Execute Usage Group usage rules:

1. Create the usage group to which the rule will belong.
2. Create the usage group that the rule will reference
3. Create the usage rule referencing the group created in the previous step

Creating Number Factors

Use the following procedure to create number factors used in usage rules.

1. Create the Characteristic Type and Values to be used by the factor that will be referenced by the rule.
2. Create the Characteristic Source Algorithm to be used by the factor that will be referenced by the rule.
3. Create the Factor that will be referenced by the rule.
4. Create the Factor Values for the factor, each referencing an effective-dated characteristic value/value pairings.

Chapter 11

TOU Maps and Dynamic Options

This chapter provides descriptions of time of use (TOU) maps and dynamic options. This chapter includes:

- **Understanding Time of Use, TOU Maps, and Dynamic Options**
- **TOU Maps in Detail**
- **Dynamic Options in Detail**
- **Dynamic Option Events in Detail**
- **Configuring TOU Maps and Dynamic Options**

Understanding Time of Use, TOU Maps, and Dynamic Options

This section describes time of use, TOU maps, dynamic options, and dynamic option events, and how they are used in Oracle Utilities meter data framework and Oracle Utilities Meter Data Management.

Time of Use

A type of rate used at many utilities is known as a "time of use" rate. For these rates, the price is based on when the usage occurs. For example, prices are typically higher during peak consumption periods (commonly referred to as "On Peak" periods) to encourage lower use during these times. The time range when a given price is applicable is known as a time-of-use (TOU) period.

Time of use periods can (and often do) change during the year.

For example, in the summer, TOU periods might be defined as follows:

Time of Use Period	Time Periods
On Peak	8:00 AM - 9:00 PM Monday - Friday
Off Peak	12:00 AM - 8:00 AM Monday - Friday 9:00 PM - 12:00 AM Monday - Friday 12:00 AM - 12:00 AM Saturday, Sunday, and Holidays

While winter TOU periods might be defined as follows:

Time of Use Period	Time Periods
On Peak	11:00 AM - 5:00 PM Monday - Friday
Off Peak	12:00 AM - 11:00 AM Monday - Friday 5:00 PM - 12:00 AM Monday - Friday 12:00 AM - 12:00 AM Saturday, Sunday, and Holidays

This is an example of a "simple" mass-market time-of-use schedule; commercial and industrial schedules can be much more complex.

Recording Usage for Time of Use Periods

Generally speaking, there are two different types of meters that can be installed at service points that participate in TOU rates:

- Scalar meters with dedicated registers for each TOU period. For this type of meter, the scalar measurements are the TOU consumption and no special derivation is necessary.
- Interval meters that measure how much is consumed each interval. For this type of meter, the interval measurements are used to derive the consumption in each TOU period.

TOU Periods are Used by the User Interface and Usage Calculation Process

While the requirement to aggregate interval consumption into TOU periods is the principal reason behind TOU maps, the system has many zones that display interval consumption using TOU periods.

The TOU concepts described in this chapter are also used when bill determinants are aggregated into TOU periods as part of the usage calculation process.

TOU Maps

Interval consumption is mapped into time-of-use periods using a TOU map. TOU maps define the TOU periods for each interval. TOU map data is structured in a format very similar to final measurement data, with specified intervals of time each designated a TOU period (instead of a measurement value). For example, the table below lists a set of TOU map data and corresponding final measurements for a period of 4 hours on January 1, 2010.

TOU Map: 463524764		**Measuring Component ID: 31425364**	
Date/Time	**TOU Code**	**Date/Time**	**Final Consumption**
1-Jan-10 3:00pm	ONPEAK	1-Jan-10 3:00pm	14.678
1-Jan-10 4:00pm	ONPEAK	1-Jan-10 4:00pm	29.12
1-Jan-10 5:00pm	ONPEAK	1-Jan-10 5:00pm	12.12
1-Jan-10 6:00pm	OFFPEAK	1-Jan-10 6:00pm	17.16

Any type of final measurement can be mapped to a TOU period, including derived quantities,

TOU Periods and TOU Groups

Some utilities use many TOU periods. For example, a TOU rate might have different prices depending on:

- The season of the year (winter, spring, summer, fall)
- The time of day (peak, off peak, shoulder)
- The actual temperature as compared to the historic average (normal, hotter, colder)

This rate would require 24 TOU periods (4 seasons * 3 time of days * 3 temperature bands). In contrast to this, another rate might have prices that differ only depending on the time of day (peak, off peak, shoulder). This rate would requires only 3 TOU periods.

To help in tracking and maintaining TOU periods, you can create TOU groups, which define the TOU periods that can be used on a TOU map.

When TOU data is created for a TOU map, only TOU periods defined on a specified TOU group can be specified.

TOU Map Templates

Every TOU map references a TOU map template that defines the rules for generating TOU data from that TOU map. Specifically, TOU map templates define:

- The TOU group (defines the valid TOU periods for the template) used for the TOU map
- The default TOU period used for periods not explicitly defined. (This means you don't have to specify dates and times for all periods. For example, if your default TOU period is "Off Peak" you only need to define dates and days and times for On Peak or other TOU periods.)
- The specific date ranges, days of the week, and time periods designated for each TOU period.

The system periodically generates TOU map data for TOU maps by interpreting the rules defined template.

Holidays

Many utilities categorize consumption on holidays differently than on the day of week on which the holiday falls. For example, holiday consumption might be categorized as Off-Peak regardless of the day it falls on. TOU map templates can define rules for different TOU periods for holidays by specifying the following:

- A Work Calendar that defines when holidays start and end
- Either:
  - A Holiday TOU period for consumption on holiday
  - A Holiday TOU Map Template that defines the TOU codes to use for different times in the year

TOU Map Template Interval Size

TOU map templates can also specify an interval size (in seconds-per-interval, or SPI). This value specifies the duration of the individual TOU map data records, and also controls the values allowed in the Start and End Times. For example, if a TOU map template sets the interval size at 15 minutes, Start and End times must be in units of the interval size (10:00, 10:15, 10:30, etc.).

A TOU map template can be used to generate TOU map data for TOU maps whose SPI is divisible by the template's SPI. For example, a 60 minute template can be used to generate TOU data for TOU maps with SPIs of 60 minutes, 15 minutes, 5 minutes, etc. This means separate map templates are not needed for every SPI.

TOU Map Types

TOU map types define attributes shared by TOU maps of a given type. Among these are the business object used for TOU maps of the given type, as well as interval size and TOU map templates.

TOU Map Type Interval Size

As noted in the **Chapter 4**: **Devices, Measuring Components, and, Device Configurations**, a measuring component's measuring component type defines its interval size (or SPI), which in turn controls the times on the measuring component's measurements.

In a similar way, every TOU map references a TOU map type that defines its interval size (and other properties). This SPI controls the times on the TOU map's TOU data.

The SPI of a TOU map must divide evenly into the SPI of any measuring component that uses the map (because the system joins the date/time of the measurement to the date/time of the TOU data). This means that it is possible to use a 15 minute TOU map with a 60 minute measuring component. However, it is not OK to have a 60 minute TOU map used with a 15 minute measuring component because the join will miss 3 out of 4 measurements.

Default and Override TOU Map Templates

While most TOU maps will use the TOU map template defined on the TOU map type, TOU maps also support a fallback/override pattern used in other areas of the system.

- A TOU map's TOU map type defines the default (or "fallback") TOU map template that's used to generate its TOU data.
- A TOU map's type defines the TOU map templates that can be referenced on individual TOU maps to override the "fallback" template.
- An individual TOU map can have an override template. If the TOU map doesn't have an override template, the fallback template defined on the TOU map type is used to generate the map's TOU data.

Dynamic Options and Dynamic Option Events

There are circumstances and conditions during which the rules for creating TOU map data might need to be calculated differently than according to the utility's standard rules. Examples of this might include critical peak periods, curtailment requests, or demand response events. During these types of events, the TOU rules defined for a TOU map must be overridden. This is done through the use of dynamic options, and dynamic option events.

Dynamic options define specific types of events which can impact how TOU map data is generated. Using the examples listed above, you might create dynamic options such as the following:

- Critical Peak Period (CPP)
- Curtailment (CURTAIL)
- Demand Response Event (DR_EVENT)

Dynamic option events define the specific periods of time during which a dynamic option is in effect. For example, if the utility identifies the period between 10:00 AM and 2:00 PM on August 2 as a critical peak period, a dynamic option event for this might look like this:

- **Dynamic Option ID**: Critical Peak Period
- **Start Date/Time**: 08/02/2010 10:00 AM
- **Stop Date/Time**: 08/02/2010 2:00 PM

A dynamic option may have many dynamic option events over time.

Dynamic Options and TOU Maps

To apply a dynamic option (and one or more of its related dynamic option events), you reference the dynamic option on a TOU map, along with a corresponding "dynamic" TOU map to be used during the dynamic option event.

To continue the example from above, for Critical Peak Periods you might add a new TOU period (called "Critical Peak"), and create a new TOU map that is the same as your standard TOU map, but that also includes your Critical Peak TOU period. For example, in the summer your new set of TOU periods might be defined as follows:

Time of Use Period	Time Periods
Critical Peak	10:00 AM - 2:00 PM Monday - Friday
On Peak	8:00 AM - 9:00 PM Monday - Friday
Off Peak	12:00 AM - 8:00 AM Monday - Friday 9:00 PM - 12:00 AM Monday - Friday 12:00 AM - 12:00 AM Saturday, Sunday, and Holidays

If a dynamic option event occurs during a usage transaction period, the standard TOU map is overridden with the "dynamic" TOU map. In our example, a usage transaction period that includes August 2010 would use the dynamic TOU map when generating TOU map data for the time between 10:00 AM and 2:00 PM on August 2, but would use its standard TOU map for the rest of the month.

TOU Maps in Detail

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

- A description of the D1-TOUMAP maintenance object
- Lists of the base package TOU map business objects, including "lite" business objects
- Details concerning TOU map-specific configuration options
- A sample TOU map business object (D2-TOUMap)

Maintenance Object - D1-TOUMAP

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

Option/Field	Description
Maintenance Object	D1-TOUMAP
Description	Time of Use Map
Service Name	D1-TOUMAP (Time of Use Maintenance)
Tables	• D1_TOU_MAP (Time of Use Map) - Primary • D1_TOU_MAP_CHAR (Time of Use Map Characteristics) - Child • D1_TOU_MAP_DYN_OPT (Time of Use Map Dynamic Option) - Child • D1_TOU_MAP_L (Time of Use Map Language) - Child • D1_TOU_MAP_LOG (Time of Use Map Log) - Child • D1_TOU_MAP_LOG_PARM (Time of Use Map Log Parameter) - Child

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

Meter Data Management Base Package TOU Map Business Objects

The Oracle Utilities Meter Data Management base package includes the following TOU map business objects:

Business Object Name	Description
D2-TOUMap	Time of Use Map Instances of this business object represent individual time of use maps defined in the system.

The base package includes the following "lite" service point business objects:

Business Object Name	Description
D2-TOUMapLite	Time of Use Map LITE

Business Object Name	Description
D2-TOUMapParentLITE	Time of Use Map Parent LITE

Configuration Options

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

Business Object Options

TOU map business objects can make use of the following business object options:

- **TOU Map Data Business Object**: This option defines the business object used for TOU map data created by TOU maps created from this business object.

System Events

TOU map business objects can make use of the following system events:

- **TOU Map (BO) - Create TOU Map Data**: This system event defines the algorithm used for create TOU map data for TOU maps created from this business object.

Example TOU Map - D2-TOUMap

The table below lists the details of the D2-TOUMap TOU map business object.

Option	Description
Business Object	D2-TOUMap
Description	Time of Use Map
Maintenance Object	D1-TOUMAP (Time of Use Map)
Application Service	D1-TOUMAPBOAS (TOU Map BO)
Instance Control	Allow New Instances
Options	• TOU Map Data Business Object: D2-TOUMapData (TOU Map Data) • Display UI Map: D2-TOUMapDisplay (TOU Map - Display) • Portal Navigation Option: d1toumapTabMenu (TOU Map) • Display Map Service Script: D2-TOUMapDis (TOU Map Display) • Maintenance UI Map: D2-TOUMapMaint (TOU Map - Maintenance)
Algorithms	• TOU Map (BO) - Create TOU Map Data: D2-CRETMD-CT (Create TOU Map Data) • Pre-Processing: D2-DEL-TOUMD (Delete TOU Map Data) • Validation: D2-TOUMAP-VL (TOU Map Validation)
Lifecycle	• Active (Initial) • Inactive (Final)

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

Dynamic Options in Detail

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

- A description of the D1-DOP maintenance object
- Lists of the base package dynamic option business objects
- A sample dynamic option business object (D2-DynamicOption)

Maintenance Object - D1-DOP

Dynamic option business objects use the D1-DOP maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-DOP
Description	Dynamic Option
Service Name	D1-DOP (Dynamic Option Maintenance)
Tables	• D1_DYN_OPT (Dynamic Option) - Primary • D1_DYN_OPT_CHAR (Dynamic Option Characteristics) - Child • D1_DYN_OPT_L (Dynamic Option Language) - Child • D1_DYN_OPT_LOG (Dynamic Option Log) - Child • D1_DYN_OPT_LOG_PARM (Dynamic Option Log Parameter) - Child

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

Meter Data Management Base Package Dynamic Option Business Objects

The base package includes the following dynamic option business objects:

Business Object Name	Description
D2-DynamicOption	Dynamic Option Instances of this business object represent individual dynamic options defined in the system.

Example Dynamic Option - D2-DynamicOption

The table below lists the details of the D2-DynamicOption dynamic option business object.

Option	Description
Business Object	D2-DynamicOption
Description	Dynamic Option
Maintenance Object	D1-DOP (Dynamic Option)
Application Service	D1-DOPBOAS (Dynamic Option BO)
Instance Control	Allow New Instances
Options	• Related Transaction BO: D2-DynamicOptionEvent (Dynamic Option Event) • Display UI Map: D2-DyOpDisplay (Dynamic Option - Display) • Portal Navigation Option: d1dyopmTabMenu (Dynamic Option) • Display Map Service Script: D2-DynOpSt (Dynamic Option Status Description) • Maintenance UI Map: D2-DyOpMaint (Dynamic Option - Maintenance)
Algorithms	• Information: D2-DOP-INFO (Dynamic Option Information)
Lifecycle	• Active (Initial)

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

Dynamic Option Events in Detail

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

- A description of the D1-DOPEVT maintenance object
- Lists of the base package dynamic option event business objects
- A sample dynamic option event business object (D2-DynamicOptionEvent)

Maintenance Object - D1-DOPEVT

Dynamic option business objects use the D1-DOPEVT maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-DOPEVT
Description	Dynamic Option Event
Service Name	D1-DOPEVT (Dynamic Option Event Maintenance)
Tables	• D1_DYN_OPT_EVENT (Dynamic Option Event) - Primary • D1_DYN_OPT_EVENT_CHAR (Dynamic Option Event Characteristics) - Child • D1_DYN_OPT_EVENT_LOG (Dynamic Option Event Log) - Child • D1_DYN_OPT_EVENT_LOG_PARM (Dynamic Option Event Log Parameter) - Child

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

Meter Data Management Base Package Dynamic Option Event Business Objects

The Oracle Utilities Meter Data Management base package includes the following dynamic option business objects:

Business Object Name	Description
D2-DynamicOptionEvent	Dynamic Option Event Instances of this business object represent individual dynamic option events defined in the system.

Example Dynamic Option - D2-DynamicOptionEvent

The table below lists the details of the D2-DynamicOptionEvent dynamic option event business object.

Option	Description
Business Object	D2-DynamicOptionEvent
Description	Dynamic Option Event
Maintenance Object	D1-DOPEVT (Dynamic Option)
Application Service	D1-DOPEVTBOAS (Dynamic Option Event BO)
Instance Control	Allow New Instances
Options	• Display UI Map: D2-DyOpEDisplay (Dynamic Option Event - Display) • Portal Navigation Option: d1dyopemTabMenu (Dynamic Option Event) • Pre-Processing Service Script: D2-DyOpEPre (Dynamic Option Event Preprocessing Service Script) • Display Map Service Script: D2-DyOpEStat (Dynamic Option Event Status Description) • Maintenance UI Map: D2-DyOpEMaint (Dynamic Option Event - Maintenance)
Algorithms	• Information: D2-DOPEVTINF (Dynamic Option Event Information) • Pre-Processing: D2-DYOPEPR (Shift Dynamic Option Event Date/Times into Standard Time)
Lifecycle	• Frozen (Initial)

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

Configuring TOU Maps and Dynamic Options

This section provides high-level overviews of the steps involved in configuring custom TOU maps, dynamic options, and dynamic option events. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

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

Configuring Custom TOU Maps

Configuring custom TOU maps involves the following steps:

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

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

   - TOU Map Data Business Object

   System Events: Create algorithms for the following system events:

   - TOU Map (BO) - Create TOU Map Data
3. Create your TOU map business objects, referencing the configuration objects created above as appropriate.
4. Set up admin records that define the TOU map types you will use in your implementation.

Configuring Custom Dynamic Options

Configuring custom dynamic options involves the following steps:

1. Design the dynamic option business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom dynamic option-related configuration objects required for your business objects.
3. Create your dynamic option business objects, referencing the configuration objects created above as appropriate.

Configuring Custom Dynamic Option Events

Configuring custom dynamic option events involves the following steps:

1. Design the dynamic option event business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom dynamic option event-related configuration objects required for your business objects.
3. Create your dynamic option event business objects, referencing the configuration objects created above as appropriate.

Chapter 12

Usage Transactions

This chapter provides descriptions of usage transactions, including:

- **Understanding Usage Transactions**
- **Usage Transactions In Detail**
- **Configuring Usage Transactions**

Understanding Usage Transactions

This section describes usage transactions and their role in the usage calculation process.

Usage Transactions Overview

Usage transactions are records of bill determinant calculations for a usage subscription. Attributes used to define usage transactions can include the following:

- **Usage Subscription**: The usage subscription for which the usage transaction was created.
- **Previous Usage Transaction**: The previous usage transaction for the referenced usage subscription.
- **Usage External ID**: An ID used by external systems used to identify the usage transaction. When Oracle Utilities Meter Data Management is integrated with Oracle Utilities Customer Care and Billing, Oracle Utilities Customer Care and Billing can send this to help identify the previous usage transaction.
- **Status**: the current state of the usage transaction
- **Start and End Date and Times**: The start and end times that define the time period for the usage transaction.
- **Usage Group**: The usage group (or groups) used to calculate usage for the usage transaction.
- **Bill Condition**: The status of the bill associated with the usage transaction (if applicable), used for informational purposes only. Valid values include Initial, Interim, and Closing.
- **Automated Retry**: A flag that indicates if creation of the usage transaction should be automatically retried in the event of an error.
- **Next Retry Date/Time**: The date and time of the next attempt to create the usage transaction (applicable only if the **Automated Retry** flag is set to "Yes").
- **Retry Until Date/Time**: The date time until which the system will attempt to retry creating the usage transaction in the event of an error (applicable only if the **Automated Retry** flag is set to "Yes").
- **Sub Usage Transactions Exist**: A flag that indicates (Yes or No) if the usage transaction is a "parent" transaction and if sub-usage transaction exist for the usage transaction.
- **Defer Calculation**: A flag that indicates (Yes or No) if calculation of the usage transaction should be deferred.
- **Trace**: Indicates if tracing is on for the usage transaction.
- **Next Scheduled Read Date**: The next scheduled read date for the device from which the measurements used to create the usage transaction came.
- **Interval Measuring Component**: Start and End dates and times for the interval measuring component that created the measurements used in calculating usage for the usage transaction (if applicable).
- **Scalar Measuring Component**: Start and End dates and times (and other details) for the scalar measuring component that created the measurements used in calculating usage for the usage transaction (if applicable).
- **Date Break**: One or more date breaks for the usage transaction. Date breaks are used to break up a usage period into sub-periods based on the dates on which rate changes took place for the service point (and its related account).
- **Issues**: A list of issues or exceptions related to the usage transaction
- **Scalar Detail**: Details of scalar measurements used in calculating for the usage transaction

- **Usage Period**: A list of the service quantities calculated for the usage transaction, including the following:
    - UOM, TOU, SQI (as applicable)
    - Quantity
    - Service Point
    - SQ (Service Quantity) Type
    - Measuring Component
    - TOU Map (if applicable)
    - Usage Group
    - Usage Rule

Usage Transactions - A Closer Look

This section provides more details concerning some specific aspects of usage transactions.

Calculation Period

Usage transaction requests must specify the date range for the usage transaction. This date range is referred to as the calculation period for the usage transaction. The dates that define the calculation period are specified by a subscribing system when it requests a usage transaction.

Date Breaks

As noted above, date breaks are used to break up a usage period into sub-periods based on the dates on which rate changes took place for the service point. For example, suppose a subscribing system requests usage for the month of January. The customer for this request has an interval meter, and the customer's usage is calculated by applying a TOU map to their interval consumption. The subscribing system detects that the customer's rate changed in the middle of January (January 16) and wants the TOU consumption calculated in two "chunks" (before and after the rate change). Because the customer has an interval meter, the exact consumption amounts before and after the rate change can be precisely calculated (as opposed to calculating each period's amount by dividing the total usage by the number of days in each period).

One approach to this situation would be for the subscribing system to request two usage transactions (where each has the desired date range). Another approach is for the subscribing system to request a single usage transaction with date breaks that define the date ranges before and after the rate change. To continue the above example, a usage transaction could be created with a date break on January 16.

If a usage transaction has date break(s), the usage calculation engine segregates the usage into multiple usage periods based on the date breaks. If there are no date breaks in the usage transaction, a single usage period is created for the entire calculation period.

As of the v2.0.0 release, the only way a usage transaction can have date breaks is if the subscribing system supplies these when it requests usage (only the subscribing system knows if and when its prices and pricing rules change during the billing period). If an implementation has additional criteria that causes date breaks, these criteria can be easily added to the usage transaction's business rules.

Service Quantities

Every usage period created for a usage transaction contains one or more service quantities.

Service quantities are calculated by the usage group’s rules specified for the usage subscription from which the usage transaction is created. For example, a usage transaction might have service quantities calculated by a single rule that applies a TOU map to the kWh channel on the device configuration installed at the service point.

Each service quantity lists details about that quantity, including a UOM (or TOU or SQI as appropriate) and a quantity. In addition, each service quantity also references the source (measuring component) of the quantities (for audit purposes).

The base-package usage rules can calculate different types of service quantities. For example, a usage group with two rules might create the following types of service quantities:

- One that retrieves and snapshots the interval for the kWh channel
- One that applies a TOU map to the channel and saves the results

> **Performance Note**: the base-package rule that snapshots intervals can create a large amount of data. This should only be used if the subscribing system requires the intervals. For example, you should not snapshot intervals on a usage transaction for audit purposes.

Service Quantities for Scalar Usage

Usage transactions and service quantities for scalar usage differ in some ways from those created from interval usage. For example, suppose a subscribing system requests usage for the month of January for a customer with a scalar meter that has been exchanged mid-month. In this case, the customer's usage is calculated by finding the scalar readings in the requested period (including all meter exchanges), and the subscribing system requires a record of all scalar readings AND a total of their consumption

Scalar rules cannot use date breaks because the system doesn't store interval values so it cannot accurately compute the amount in each period. In this case, scalar usage rules retrieve scalar readings for the service points linked to the usage subscription, and then creates a single usage period for the usage transaction's entire calculation period. The individual scalar readings are captured as "scalar details" in the usage transaction.

To continue the example above, there would be 2 entries in the Scalar Details, one for each meter, and a single usage period that contains the total consumption for the entire calculation period. The tables below illustrate what this might look like.

Scalar Details:

Final Usage	Usage Group	Measured Usage UOM/TOU/SQI	Service Point	Measuring Component	Start Reading	Stop Reading
202 BTU	Residential / Retrieve Scalar	200 CCF	39191912312 (18 Main St)	8382821921 (CCF register)	1-Jan-10 3pm 10000	15-Jan-10 6pm 10200
303 BTU	Residential / Retrieve Scalar	300 CCF	39191912312 (18 Main St)	8382821921 (CCF register)	15-Jan-10 6pm 10200	30-Jan-10 4pm 10500

Usage Period:

Date/Time Range
1-Jan-10 3:00pm to 30-Jan-10 4:00pm

UOM/TOU/SQ	Usage
BTU	505 BTU

This example also illustrates how a usage rule can convert a Measured UOM (CCF) into a Final UOM (therms).

Usage Transaction Corrections Based on Initial Measurement

In some situations, existing usage transactions can be impacted by incoming initial measurement data if the start and stop date/time of the initial measurement overlaps with the date range of the usage transaction. The "Finalized" state of the initial measurement business objects have been updated to invoke a new algorithm (D1-TRAN-UT) that creates a "Usage Transaction Correction Processor" activity for each usage transaction that may have been impacted by an initial measurement.

This algorithm retrieves all usage transactions that overlap with the current initial measurement's Start and End Date/Times and that are linked to a usage subscription associated to a service point at which the device is installed for the initial measurement's measuring component The usage transaction selection filters out the sub usage transactions and processes only "parent" usage transactions.

For each usage transaction retrieved, the algorithm checks to see whether there is a Usage Transaction Correction Processor activity (of the same business object as referenced in the algorithm's soft parameter) in a non-final state that references the current usage transaction. If so, the current usage transaction is ignored and the next usage transaction is processed among those impacted by the current initial measurement.

If there is no Usage Transaction Correction Processor activity existing for the usage transaction (or if the only Usage Transaction Correction Processors existing for the usage transaction are in a final state), a new Usage Transaction Correction Processor activity is created using the business object referenced in the soft parameter (If the soft parameter is left blank or an active Activity Type cannot be found for the business object referenced in the soft parameter, no Usage Transaction Correction Processor activity will be created).

Usage Transaction Exceptions

Usage rules can both calculate usage and validate that the calculated usage is reasonable. The usage calculation process creates an exception for each problem encountered. Multiple exceptions can be created when a usage transaction is subject to validation. This allows users to see all of the problems detected by the usage calculate process.

> **Note**: Usage exceptions are held within the usage transaction (in its CLOB). This contrast to VEE exceptions is intentional as usage transaction exceptions should be very rare whereas VEE exceptions are relatively common.

Exception Categories

Similar to VEE exceptions, there are three categories, or severities of usage exceptions:

- **Info**: Used to highlight something interesting, but not sufficient to cause the usage calculation to be put into the exception state.
- **Issue**: Used to report a problem that will prevent the usage transaction from being finalized. Multiple "issue exceptions" can be created during usage calculation. If at least one issue exists after all rules have been applied, the usage transaction is transitioned to the exception state.
- **Terminate**: Used to report a severe issue that will cause the usage calculation process to stop and the usage transaction to be transitioned immediately to the exception state. Only one terminate exception can be issued (as the first one causes usage calculation to stop).

Exceptions and To Do Entries

In addition to exceptions, usage processing can also trigger the creation of To Do Entries related to failed validations.

If Issue or terminate exceptions exist for an initial measurement, a To Do Entry is created when the usage transaction is transitioned to the Exception state. The To Do Type and default To Do Role of this To Do Entry are defined on the Enter system event for the Exception state of the business object used to define the usage transaction.

To Do Entries created in this way can be routed to different roles depending on the exception's message category and number (using the To Do Type's Message Overrides tab).

Available Actions for Usage Transactions with Exceptions

Users have a number of options for dealing with usage transactions with exceptions.

- After correcting the cause of the issues that triggered the exceptions, a user can re-calculate the usage transaction.
- A user can discard the usage transaction.
- A user can manually complete the usage transaction. This sends the usage transaction to the subscribing system “as is”
- A user can edit the Post VEE quantities (if necessary) and manually complete the initial measurement. This will cause final measurements to be created using the contents of the Post VEE quantities.

  Note: No VEE processing is performed on manually completed initial measurement data.

Regardless of the action taken by the user, the system will complete any open To Do Entries that created when the usage transaction entered the Exception state.

Creating Usage Transactions from External Systems

When Oracle Utilities Meter Data Management is integrated with a customer information system (such as Oracle Utilities Customer Care and Billing), usage transactions can be created via a request from the customer information system.

To invoke a usage transaction request, the external system must invoke an XAI Inbound Service mapped to the Usage Transaction Seeder (D2-UsgTranSeeder) business object. This business object does the following:

- Determines the usage subscription ID based on an external usage subscription ID

  This processing is performed via the Determine Usage Subscription ID (D2-DETUSID) Pre-Processing algorithm.

- Determines the appropriate usage transaction business object to create.

  This processing is performed via the Determine Usage Transactions Business Object (D2-DETUTBO) Pre-Processing algorithm.

  This algorithm uses the “How To Create Usage Subscription Related Information” (D2-HowToCreateUSInformation) processing method defined for the “Usage Transaction Creation” processing role on the service provider that represents the external system.

In addition to the Usage Transaction Seeder business object, the Oracle Utilities Meter Data Management base package also includes the following XAI configuration options to support creating usage transactions from external systems:

- **XAI Inbound Service**: D2-UsageTransactionRequestInbound (D264745327)

  This inbound service is mapped to the Usage Transaction Seeder (D2-UsgTranSeeder) business object.

Usage Transactions In Detail

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

- A description of the D1-USAGETRANS maintenance object
- Lists of the base package usage transaction business objects, including "lite" business objects
- A sample usage transaction business object (D2-UsageTransaction)

Maintenance Object - D1-USAGETRAN

Usage transaction business objects use the D1-US maintenance object. The table below outlines some of the details of this maintenance object

Option/Field	Description
Maintenance Object	D1-USAGETRAN
Description	Usage Transaction
Service Name	D1-USAGETRAN (Usage Transaction Maintenance)
Tables	• D1_USAGE (Usage) - Primary • D1_USAGE_CHAR (Usage Characteristics) - Child • D1_USAGE_LOG (Usage Log) - Child • D1_USAGE_LOG_PARM (Usage Log Parameters) - Child • D1_USAGE_PERIOD (Usage Period) - Child • D1_USAGE_PERIOD_SQ (Usage Period Service Quantity) - Child • D1_USAGE_REL (Usage Relationship) - Child • D1_USAGE_SCALAR_DTL (Usage Scalar Detail) - Child
Algorithms	• Information: D1-USAGEINFO (Usage Transaction Information)

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

Meter Data Management Base Package Usage Transaction Business Objects

The Oracle Utilities Meter Data Management base package includes the following usage transaction business objects:

Business Object Name	Description
D2-SubUsageTransaction	Sub Usage Transaction Instances of this business object represent individual sub usage transactions created in the system.
D2-UsageTransaction	Usage Transaction Instances of this business object represent individual usage transactions created in the system.

The base package includes the following “lite” usage transaction business objects:

Business Object Name	Description
D1-UsageTransactionParentLite	Usage Transaction Parent LITE
D2-UsageIntervalDataLite	Usage Transaction Interval Data LITE
D2-UsageTranExceptionsLITE	Usage Transaction Exceptions LITE
D2-UsageTranScalarDetailsLite	Usage Transaction Scalar Details LITE
D2-UsageTransactionDateBreaks	Usage Transaction Date Breaks LITE
D2-UsageTransactionExceptions	Usage Transaction Exceptions LITE
D2-UsageTransactionMini1	Usage Group Transaction Processing
D2-UsageTransactionStatusLite	Usage Transaction Status LITE
D2-UsageTransacitonTrace	Usage Transaction Trace LITE
D2-UsageTransactionUsagePeriod	Usage Transaction Usage Periods LITE
D2-UsgTranChkSubLITE	Usage Transaction Check Sub LITE
D2-UsgTranDeferLITE	Usage Transaction Defer LITE
D2-UsgTranRetryLITE	Usage Transaction Retry LITE
D2-UsgTranSkipDetailsLite	Usage Transaction Skip Details Lite
D2-UsgTranSubLITE	Usage Transaction Sub LITE

The base package includes the following additional usage transaction business objects:

Business Object Name	Description
D2-UsgTranSeeder	Usage Transaction Seeder (used to determine the usage transaction business object to use when creating new usage transactions)

Example Usage Transaction - D2-UsageTransaction

The table below lists the details of the D2-UsageSubscription usage transaction business object.

Option/Field	Description
Business Object	D2-UsageTransaction
Description	Usage Transaction
Maintenance Object	D1-USAGETRAN (Usage transaction)
Application Service	D2-USAGETRANBOAS (Usage transaction BO)
Instance Control	Allow New Instances
Options	• Display UI Map: D2-UsageTranDisplay (Usage Transaction - Display) • Portal Navigation Option: d1utnmTabMenu (Usage Transaction) • Pre-Processing Service Script: D2-UsgTraPre (Usage Transaction Pre-Processing) • Display Map Service Script: D2-UsgRetDts (Usage Retrieve Details for Display) • Maintenance UI Map: D2-UsageTranMaint (Usage Transaction - Maintenance)
Algorithms	• Validation: D2-DEL-ACT (Delete Validation) • Validation: D2-SQTYP-VAL (Validate Usage Transaction Service Quantity Entries) • Validation: D2-INSCP-VAL (Interval and Scalar Period Validation)
Lifecycle	• Pending (Initial) • Calculate (Interim) • Prorated (Final) • Approval In Progress (Interim) • Issue Detected (Interim) • Approve (Interim) • Sent (Final) • Subsequent Correction (Final) • Discarded (Final) • Calculation Deferred (Interim) • Calculate in Progress (Interim)

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

Configuring Usage Transactions

This section provides high-level overviews of the steps involved in configuring custom usage transactions. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

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

Configuring Custom Usage Transactions

Configuring custom usage transactions involves the following steps:

1. Design the usage transaction business objects you will need to create for your implementation, including the data and processing required for each.
2. Create the custom usage transaction-related configuration objects required for your business objects.
3. Create your usage transaction business objects, referencing the configuration objects created above as appropriate.

Chapter 13

Aggregations

This chapter describes aggregations and how they are performed in Oracle Utilities Meter Data Management, including:

- **Understanding Aggregations**
- **Configuring Aggregations**

Understanding Aggregations

This section describes aggregations and how they are created and calculated in Oracle Utilities Meter Data Management.

The term aggregation refers to the summarization of measurements and statistical values from a group of related measuring components. Aggregated values are used primarily for analysis. For example, an electric utility may want to see aggregated consumption for each transformer within its service territory, where a transformer is uniquely identified by 3 values:

- The Transformer itself,
- The Feeder associated with the Transformer, and
- The Substation associated with the Feeder.

Aggregator Measuring Components and Dimensions

Aggregator measuring 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, or for each transformer within its service territory (as above). The aggregated values of an aggregator measuring component's constituent measuring components are its final measurements. Aggregator measuring components do NOT have initial measurements.

Every type of aggregation has one or more dimensions that define how to gather the constituent measuring components from which measurements are aggregated. For example, the transformer aggregation referenced above has 3 dimensions: Substation / Feeder / Transformer.

An aggregator measuring component must exist for every distinct combination of dimensions. For example, the transformer aggregation requires a separate measuring component for every distinct combination of substation / feeder / transformer found on the electric service points.

Any combination of dimensions is possible. The prior example illustrated an aggregator measuring component whose dimensions are all derived from the same type of object, in this case, the substation, feeder, and transformer defined on electric service points.

Aggregator measuring components can exist for dimensions derived from any data (including admin objects). For example:

- A Postal Code / Measuring Component Type aggregation has a separate aggregator measuring component for every combination of postal code and measuring component type
    - Postal code is an attribute of each service point
    - Measuring Component Type is an attribute of each measuring component
- A Rate / Postal Code aggregation has a separate aggregator measuring component for every combination of rate and postal code
    - Rate is an attribute of each usage subscription
    - Postal code is an attribute of each service point

The base package and demonstration data include several examples of different dimensional combinations. Implementations can introduce others based on their requirements.

Types of Aggregated Data

Any of the quantities on final measurements can be aggregated.

Interval measuring components of different magnitudes, such as KW vs MW, SPIs (15 minute vs 60 minute) and, in some cases, units of measure can be aggregated together.

Note: Not all units of measure can be aggregated together. Only those that share a common Base Unit of Measure can be aggregated together. For example, KW can be converted to kWh and vice versa, but BTU measurements cannot be converted to KW.

This requires each constituent measuring component to be converted into a common UOM and SPI. Converting different UOMs to a common UOM uses the Base Unit of Measure and Magnitude attributes. Converting differing SPIs to a common SPI involves scaling the interval values up or down accordingly.

Both scalar and interval measuring components can also be aggregated together. This requires the scalar consumption to be intervalized into the aggregator measuring component's UOM and SPI prior to aggregation.

Aggregating Aggregations

Aggregations can also be aggregated into higher levels. For example, the aggregator measuring components whose measurements hold aggregated consumption for each Substation / Feeder / Transformer combination can be used to aggregate consumption at each Substation / Feeder.

In this case, distinct aggregator measuring components must exist for every Substation / Feeder combination. The measurements of these aggregator measuring components are the sum of the consumption on the transformers linked to each feeder.

Note that these values can be derived from the Substation / Feeder / Transformer aggregator measuring components, meaning that it is only necessary to aggregate the transformer measurements once.

In turn, aggregator measuring components that hold aggregated consumption for each Substation / Feeder combination could be used to aggregate consumption at each Substation. Again, distinct aggregator measuring components exist for every Substation, and their measurements would represent the sum of their feeders.

Finally, total consumption for all substations could be aggregated using the same technique.

Aggregation Calculations

The system periodically aggregates consumption via batch process, using a deferred monitor on the aggregator measuring components. In addition, users can re-aggregate data in real-time if they don't wish to wait for the batch process, or if the original aggregation needs to be re-calculated due to incorrect data. Users can also create ad hoc aggregations "on the fly" and these will persist in the database in the same manner as other aggregations.

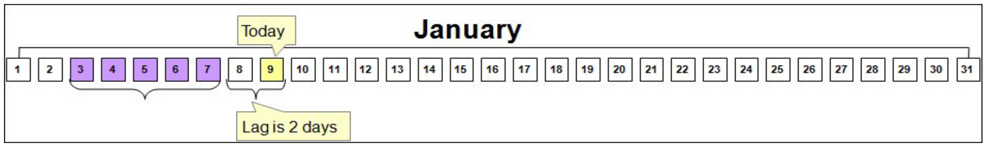
Understanding Aggregation Periods

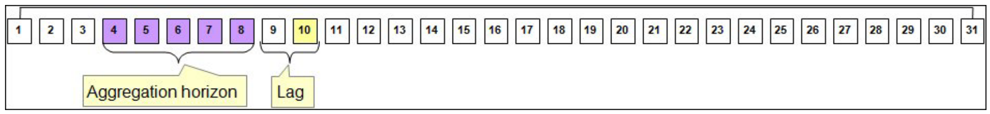
The start and end dates and times for aggregation calculations are based on the following:

- Aggregation Horizon
- Aggregation Lag
- Aggregation Cut Off Time

Whenever aggregation is performed for an aggregator measuring component, consumption is aggregated for every day in its "Aggregation Horizon." The "Aggregation Horizon" is the number of days during which there's a potential change in measurement data for one or more of the measuring components associated an aggregator measuring component.

Aggregation calculations typically lag behind the current date by a few days to give the system time to upload and perform validations and create final measurements. The amount of lag time is referred to as the "Aggregation Lag" and is the number of days between the date on which aggregation calculations are performed and the end date of the aggregation period. This defines the time period between the aggregation calculation date and the end of the aggregation horizon that serves to allow all measurements to arrive. This together with the Aggregation Horizon is used to determine the start and end dates of an aggregation period. For example, with an Aggregation Horizon of 5 and an Aggregation Lag of 2, aggregation calculations performed on January 9 would be for an aggregation period of January 3 through January 7. The next day (January 10), the aggregation period would shift to January 4 through January 8.





Aggregation is always performed through a given "through time" (such as 12:00 AM) rather than through the actual time of the aggregation calculation. This time is referred to as the “Aggregation Cut Off Time.” For example, the stop time for aggregation calculations with an Aggregation Lag of 2, and an Aggregation Cut Off Time of 10:00 PM will always be 10:00 PM 2 days prior to the date on which the calculations are performed. In the above examples, aggregations performed on January 9 would have an end date/time of 10:00 PM on January 7, and aggregations performed on January 10 would have an end date/time of 10:00 PM on January 8.

The Aggregation Horizon, Aggregation Lag, and Aggregation Cut Off Time are configured for each aggregator measuring component type.

Aggregation and Re-Aggregation

The use of the aggregation horizon means that aggregated totals for some days will be re-aggregated until those days are no longer covered by the aggregation horizon. For example, with an aggregation horizon of 5 days and an aggregation lag of 2 days, on the night of January 9 the aggregation period would be January 3 through January 7 (as in the above example).

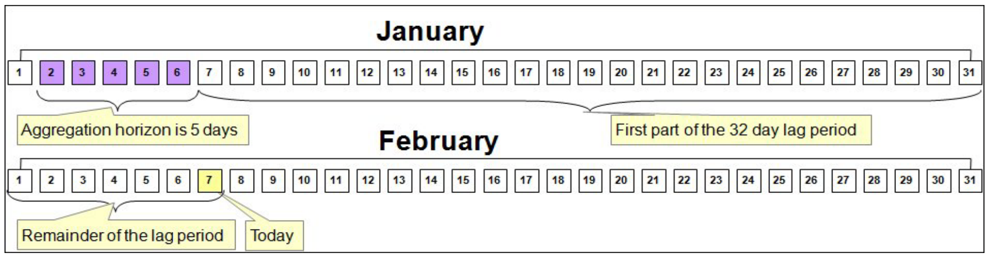
On the night of January 10, the horizon will shift 1 day (to January 4 through January 10, again as above). This means the following:

- The totals for January 3 calculated on January 9 will be untouched (January 3 now falls outside the aggregation horizon)
- The totals for January 4 through January 7 will be re-derived because corrections may have occurred (and they still fall within the aggregation horizon)
- The totals for January 8 will be calculated for the first time (because it now is within the aggregation horizon).

Manually Read Meters and Aggregation Lag

Aggregations that include manually read meters often have much longer aggregation lags than those for automatically read meters. This allows more time for manual meter readings to be imported into the system for use in aggregations.

For example, suppose a situation where manual meter reads arrive approximately 1 month after the date of the reading. In this case, it wouldn't be until a meter read upload on February 7 that the last manual reads including consumption from January 6 will exist. Since we don't want to perform aggregations for January 6 until there's a decent chance that all consumption for that date exists, an aggregation lag of 32 days ensures that the data for January 6 is in the system when the aggregation is performed on February 7.



When An Aggregator Measuring Component Contains Both Manual and AMI Measuring Components

The previous section suggests that if an aggregator measuring component contains both manually read and AMI measuring components, the aggregator measuring component should have a long lag time rather than the short time shown in the earlier example.

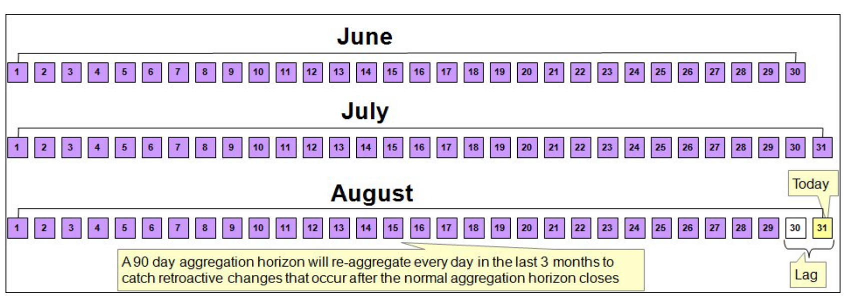
While this is one possible approach, an alternative approach could allow users to see timely aggregations of the AMI channels and only lag the manual channels. To do this they could configure the system as follows:

- Create an aggregator measuring component that only aggregates manual measuring component types (horizon: 5 days, lag: 31 days)
- Create a separate aggregator measuring component that only aggregates interval measuring component type: 5 days types (horizon: 5 days, lag: 2 days)
- Create a third aggregator measuring component that aggregates the above aggregator measuring component types (horizon: 5 days, lag 31 days)

Expanding The Aggregation Horizon Periodically

Even with a 30-ish day aggregation lag, it's possible for an account with several months of estimated consumption to have its consumption "invalidated" when real readings arrive whose dial readings are less than the estimated readings (or when major retroactive data change takes place). Another similar situation is when a user manually corrects consumption after the aggregation horizon has moved forward.

To address situations like these, the use of ad-hoc aggregations with a user-specified aggregation horizon supports the notion of periodically (such as once a month or once a week) aggregating with a long aggregation horizon (up to 90 days) to catch as many retroactive changes as possible.



A Note About Using Non-Effective Dated Dimensions

Be aware that whenever aggregation takes place, the system uses master data that exists as of that date. For example, if an aggregation is performed using non effective-dated master data attributes (for example, aggregating by measuring component type) and a measuring component's type is changed, all days in the aggregation horizon will reflect the change regardless of when the user makes the change, because the measuring component type relationship is not effective-dated.

The impact of this depends on the nature of the change:

- If the master data was wrong, this provides a way to re-aggregate the data to accurately reflect the master data.
- If the change occurred effective as of a given date (but there's no effective dated attribute), this can result in incorrect aggregation values.

Because of this, it is recommended that where possible dimensions used in aggregations be effective-dated.

Aggregator Measuring Component Types

Aggregator measuring components have measuring component types just like all measuring components. In the case of aggregator measuring components, the measuring component type controls the aggregation period (horizon, lag, and cut off time) as well as the type of measuring components that are aggregated.

Aggregation Group

These attributes are used to define the aggregation period as described above, and include:

- Aggregation Horizon (in days): the number of days during which there's a potential change in measurement data for one or more of the measuring components associated an aggregator measuring component type.
- Aggregation Lag (in days): the number of days between the date on which aggregation calculations are performed and the end date of the aggregation period.
- Aggregation Cut Off Time (a time): the end time for aggregation calculations performed for aggregator measuring components of this type. This is used to ensure a consistent end time on all aggregation horizons and this is important if an aggregator is going to aggregate other aggregators.

Valid MC Types to Aggregate

Defines the valid measuring component types that can be included for aggregation calculations for aggregator measuring components of this type. Only measuring components whose types are defined here will be included in aggregation calculations for aggregator measuring components of this type. Note that any measuring component type (aggregator, physical, scratchpad, etc.) can be defined as long as the UOM of the identified value can be converted into the UOM of the aggregator measuring component's primary measurement value.

Value Identifiers

Defines value identifiers related to the current measuring component type. Value identifiers are used to provide short descriptions for the various types of values measured by the measuring component (KWH, KW, etc.).

Aggregator measuring components can derive up to 10 values in addition to the basic value (just like other measuring components). For example, the minimum, maximum, and average values could be stored on each measurement. The way the other values are derived is controlled by an enter algorithm on the Aggregate state.

Measuring Component Attributes

In addition the attributes on an aggregator measuring component's type, aggregator measuring components also have attributes that define how aggregations are performed. These include:

One Time Aggregation

Controls if the aggregator measuring component is used for one-time aggregations (used for ad hoc aggregations).

Consumption Aggregated Through Date Time

The date and time up to which consumption has been aggregated for the measuring component, based on the aggregation horizon used during the last aggregation calculation.

Next Aggregation Horizon

Defines the horizon for the next aggregation (either entered by a user or computed by the monitor algorithm when it determines it's time to aggregate).

Creating Aggregator Measuring Components

There are two ways in which aggregator measuring components are created:

- Automatically: The system will periodically create new aggregator measuring components when new "dimensional values" are detected. For example, if a new substation is added to the system, a new aggregator measuring component will be created for the substation.
- Manually: An end-user can create an aggregator measuring component at their discretion

Automatic Creation of Aggregator Measuring Components

The system automatically creates aggregator measuring components when it detects new instances of the dimension(s) being aggregated. This activity is referred to as dimension scanning. For example, if the system is configured to aggregate individual substations, a user will NOT have to manually set up an aggregator measuring component when a new substation if referenced on a service point. Rather, the system will create the new aggregator when it detects a substation on a service point that doesn't have an aggregator measuring component.

Similarly, if is configured to aggregate distinct combinations of rate class (on usage subscription type) and postal code (on service point), the system will create an aggregator measuring component when it detects new combinations of rate class and postal code.

The periodic monitoring to ensure sufficient aggregation measuring components exist is implemented via monitoring algorithms on a related activity business object that references a corresponding measuring component type. This means that there must be a separate activity type / activity combination for every type of aggregation performed. These monitor algorithms include:

- A Monitor algorithm on the Active state that simply transitions the activity to the transitory Scan state.

  Users can manually transition the activity to the Scan state by clicking the “Scan” button on the activity.
- An Enter algorithm on the Scan state finds the distinct combinations of dimensional values for this type of aggregation and checks that an aggregator measuring component that references the activity's aggregator measuring component type exists for every instance and creates new aggregator measuring components as appropriate. The algorithm also populates the dimensional values on the new measuring components (mapped to searchable characteristics), and sets the Consumption Aggregated Through Date/Time field on the new aggregator measuring components to the current date and the Aggregation Cut Off Time on the aggregator measuring component type.

Hard-Wiring Service Types for Aggregations

If the dimensions used for aggregation do not include at least one service type-specific dimension (for example, in the case of aggregating by postal code only) AND the aggregation should not commingle measurements of different service types, it's important to:

- Declare the specific service type for the aggregation on the dimension scanner activity instance
- Declare the desired service type-specific measuring component types on the aggregator measuring component type

For example, if an electric and gas implementation aggregates consumption by postal code, they will need 2 activities: one will reference the Electric service type and the desired aggregator measuring component type, the other will reference the Gas service type and its aggregator measuring component type

The aggregator measuring components will have two dimensional attributes: postal code and service type.

Manual Creation Of Aggregator Measuring Components

Users can also create aggregator measuring component manually. This is useful when an ad hoc aggregation for an ad hoc time period is required, or when there is no activity that automatically creates instances of a given combination of dimensions.

To do this, the user simply clicks the “Add” link on the Aggregator Search zone title bar. The BPA script:

- Prompts the user to define the type of aggregation (this will present a list of measuring component types whose measuring component category is aggregator)
- Prompts the user to define the dimensional value(s), the desired aggregation time, and if this aggregation should be performed indefinitely or if it is a "one time"

Aggregation Algorithms

The base package includes an aggregation algorithm that can populate columns on aggregated final measurements with the sum, max, min, average, and count of measurements in an interval. Additional algorithms can be developed by implementations teams to support other types of aggregations. For example, imagine a scenario where a user wants to see aggregated consumption during a critical peak period and contrast this to "normal" consumption. In this case, columns on the aggregated measurement could contain values like:

- Actual consumption
- “Normal” consumption
- The count of customers who reduced consumption between 0 and 10%
- The count of customers who reduced consumption between 10 and 25%
- The count of customers who reduced consumption more than 25%
- The count of customers who increased consumption between 0 and 10%
- The count of customers who increase consumption more than 10%

This scenario would necessitate the creation of a new aggregation algorithm that could piggy back on the base package version, and would only need additional logic to calculate each customer’s "normal" consumption for each interval and populate the "counts" accordingly.

Aggregating Specific Measuring Components

The above sections describe dimension-oriented aggregations, of the means of aggregating a variety of measuring components that are related to a given set of dimensional values (such as all measuring components for a given service type in a given postal code.

However, there is another class of aggregations where the requirements call for aggregating a set of specific measuring components. In addition, the aggregation formulae can be very unique, such as (56.2% of MC1 - 45.3% of MC2) * 1.034.

Very specific aggregations of this should be implemented using usage rules, not aggregator measuring components.

Configuring Aggregations

This section provides high-level overviews of the steps involved in configuring custom aggregations. See **Configuration Process Overview** in **Chapter One** for a high-level overview of the overall configuration process.

Configuring Custom Aggregations

Configuring custom aggregations involves the following steps:

1. Create a business object for the aggregator measuring component.

   This will flatten the dimensional value(s) into searchable characteristics.

   Whether this business object is a parent or a child of another aggregator business object depends on when periodic aggregation should occur:

   - If you want the periodic aggregation to occur when another aggregation occurs, it can be a child business object (meaning that it inherits the lifecycle (and therefore the deferred monitor) of the parent)
   - If you want to schedule its periodic aggregation independently from other aggregation business objects, this must NOT be a child business object as it will require its own deferred monitor (and deferred monitors can only be defined on parent business objects)
2. Create UI maps for the aggregator business object as follows:
   - One to display the aggregator measuring component (Display)
   - One to allow user to change / add a new one (Maintenance)
3. Create an info plug-in for the aggregator business object that concatenates together its dimension types and values.
4. Create a "Find Constituent Measuring Components" algorithm and plug it on the aggregator business object.

   This will be passed the aggregator measuring component and the from and to date/times. It will insert the constituent measuring component IDs and the respective from / to date-time of each onto a temporary table.
5. Create an measuring component type instance and reference the new aggregator measuring component business object (as well as the types of constituent measuring component types that should be aggregated).
6. Create a query zone for Consumption Statistics search to allow users to find the aggregator measuring component.

Optionally:

7. Create a business object for the dimension scanner activity.

   This should be a child business object of the base package dimension scanner business object.
8. Create UI maps for the activity business object, as follows:
   - One to display the dimension scanner activity (Display)
   - One to allow users to change / add a new one (Maintenance)
9. Create an info plug-in that will describe what it scans.
10. Create an Enter algorithm on the Scan state that finds distinct combinations of the dimensional values and creates new aggregator measuring components when new ones are detected.

    You can reuse the base package deferred monitor batch control.

11. Create an activity type and reference the new dimension scanner business object.

Meter Data Management Base Package Aggregations

The Oracle Utilities Meter Data Management base package includes the following base package aggregators, delivered as aggregator measuring component business objects and measuring component types.

Business Intelligence Aggregators

The following aggregators are used by Oracle Utilities Meter Data Management Business Intelligence, and aggregate measurement data (including quantities and counts) for constituent measuring components based on the following dimensions: Postal, City, Head-End, Device Type, Usage Calculation Group, Market and Service Provider, and Service Type.

Aggregator	Description
Measurement Measured Quantity Aggregator (D2-MeasuredQuantityAggregator)	Aggregates measurement quantities and distributes the constituents' measurements across Value Identifiers based on condition codes.
Measurement Quality Count Aggregator (D2-MsrmtQualityCountAggregator)	Aggregates measurement counts and distributes the constituents' counts across Value Identifiers based on condition codes.
Measurement Timeliness Count Aggregator (D2-MsrmtTimelinessCountAggr)	Aggregates measurement counts that arrived on time or are late.
Measurement Timeliness Quantity Aggregator (D2-MsrmtTimelinessQuantityAggr)	Aggregates measurement quantities that arrived on time or are late.

Sample Aggregator - Postal and Service Type

The Oracle Utilities Meter Data Management base package also includes an example aggregation that aggregates measurement quantities for constituent measuring components based on postal code and service type dimensions. The table below outlines the types of objects used in this aggregation, based on the steps outlined above), and the specific objects for each type.

Object Type	Base Package Example
Aggregator Measuring Component Business Object (Step 1)	D2-Aggregator (Aggregator - Postal and Service Type)
Aggregator Measuring Component UI Maps (Step 2)	Display: D2-AggMCDisp (Service Type and Postal Aggregator-Display) Maintenance: D2-AggMCMaint (Service Type and Postal Aggregator-Maintenance)
Aggregator Business Object Info Algorithm (Step 3)	D2-AMC-INFO (Service Type and Postal Aggregator - Information)
Find Constituent Measuring Components Algorithm (Step 4)	D2-DET-CMC (Find Constituent Measuring Components Based on Service Type and Postal)
Measuring Component Type (Step 5)	D2-AggregatorType (Aggregator Type)

Object Type	Base Package Example
Query Zone for Consumption Statistics Portal (Step 6)	D2-AGGMCQRY (Aggregator Search)
Dimension Scanner Activity Business Object (Step 7)	D2-ActivityAggDimScanner (Aggregator Creator - Postal / Service Type)
Dimension Scanner Activity UI Maps (Step 8)	Display: D2-AggDimScannerActDisp (Aggregator DS Activity-Display) Maintenance: D2-AggDimScannerActMaint (Aggregator DS Activity-Maintenance)
Dimension Scanner Activity Business Object Info Algorithm (Step 9)	D2-ADS-INFO (Aggregator Dimension Scanner Information)
Enter Algorithm for Scan State (Step 10)	D2-CRE-AGGMC (Aggregator MC Creation for Post Code and Service Type)
Activity Type (Step 11)	D2-ActivityTypeAggDimScanner (Aggregator Dimension Scanner Activity Type)

Chapter 14

Batch Processing

This chapter describes the base package batch processes provided with Oracle Utilities meter data framework and Oracle Utilities Meter Data Management. Use the Batch Control portal and Application Viewer for more details about these batch processes. This chapter includes:

- **Meter Data Framework Batch Controls**
- **Meter Data Management Batch Controls**

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

- Oracle Utilities Meter Data Management Batch Server Administration Guide
- Oracle Utilities Application Framework Business Process Guide (Batch Jobs)
- Oracle Utilities Application Framework Administration Guide (Defining Background Processes)

Meter Data Framework Batch Controls

The table below lists the batch controls provided in the meter data framework 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-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
D1-ICERR	Inbound Communication Error - Retry
D1-IMD	IMD Monitor 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.
D1-IMDMC	IMD Monitor Unattached MCs
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.

Batch Control	Name / Description
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-OCERR	Outbound Communication Error - Retry
D1-OCWT	Outbound Communication Wait - Monitor
D1-PMCS	Process Measurement Cycle Schedule Looks for measurement cycle routes that are scheduled to be processed and populates them on the measurement cycle schedule. Once the schedule is prepared, the Process Measurement Cycle algorithm (specified on the service point business object) is then invoked for each device in every route that is ready for processing.
D1-PSPSR	Process SP / MC Schedule Route Records
D1-SMMTR	Smart Meter State Monitor Process
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-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 meter data framework 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

Meter Data Framework Batch Processing Guidelines

This section provides some guidelines around how to schedule batch processing for the batch controls used with Meter Data Framework (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 requets 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 traped, 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 measurementss 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.

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 peformed for long time periods, such as a year, generating data for all time-of-use maps for the entire following year, or it could be done more frequently, such as whenever schedules are updated via the TOU map templates.

Re-Derive Measurement Values

If certain data such as a factor value was found to be incorrect, derived measurement values might need to be re-calculated across all final measurements for a given date range. This can be performed using the "Measurement - Derive Other Values" batch control (D1-MSRMT). Given the possible volume of data impacted by this process, careful consideration should be given before performing this process.

Master Data Monitoring

The base package also provides batch processes intended to monitor service points (D1-SP), install events (D1-INEVT), and measuring components (D1-MC) on an ad-hoc basis if processing driven from the lifecycles of these objects is needed. Note that the base package contains no such processing.

Meter Data Management Batch Controls

The table below lists the batch controls provided in the Oracle Utilities Meter Data Management base package.

Batch Control	Name / Description
D2-ADS	Aggregator Dimension Scanner Monitor
D2-AGG	Aggregation Monitor
D2-SIKUS	Initial Sync Request - Resolve Keys US
D2-SILUS	Initial Sync Request - Load Data US
D2-UT	Usage Transaction Monitor Process
D2-UTCD	Usage Transaction Calculate Defer Monitor
D2-UTID	Usage Transaction Issue Detected Monitor
D2-UTSED	Usage Transaction Seeder - Error

Meter Data Management Batch Processing Guidelines

In addition to the batch processes outlined for the Meter Data Framework above, Oracle Meter Data Management can employ the following ad-hoc or periodic batch processes.

Usage Transaction processing

If Oracle Utilities Meter Data Management is integrated with a billing system (such as Oracle Utilities Customer Care and Billing) that requests bill determinants, usage transaction processing should be coordinated with that billing system's billing process. This can include:

- Reprocessing usage transaction "seeders" in error (D2-UTSED)
- Reprocessing usage transactions in error (D2-UTID)

Requests from a billing system that have been indicated as "batch requests" (such as those produced by Oracle Utilities Customer Care and Billing batch billing process) accumulate in a "calculation deferred" state to be processed specially by the “Usage Transaction Calculate Defer Monitor” batch control (D2-UTCD).

If unexpected errors occur that leave usage transactions in an unmonitored state, the “Usage Transaction Monitor” batch control (D2-UT), or one based on this batch control with parameter values tailored to any specific requirements, can be used to process those usage transactions.

Aggregation

Aggregation calculations should be run on an as needed basis. This can include:

- Scanning for new aggregation dimension (D2-ADS)

  This process is applicable if the system is configured to use aggregation dimension scanners to detect new aggregation dimensions (such as a service point referencing a new transformer for which an aggregator measuring component doesn’t currently exist)
- Performing aggregation calculations (D2-AGG)

Note that aggregation calculations should precede usage transaction processing if aggregated values serves as input to the calculation of bill determinants.

Meter Data Management Business Intelligence Batch Controls

The table below lists the batch controls used with Oracle Utilities Meter Data Management Business Intelligence.

Batch Control	Name / Description
D1-ACSDX	Activity BO Status/Reason Dim Extract
D1-ACSIL	Activity BI Status/Reason Initial Load
D1-ACTFX	Activity Accumulation Fact Extract
D1-ACTL	Activity Accumulation Fact Initial Load
D1-ADRDX	Address Dimension Extract
D1-ATYDX	Activity Type Dimension Extract
D1-ATYIL	Activity Type Dimension Initial Load
D1-DESDX	Device Evt BO Status/Reason Dim Extract
D1-DESIL	Device Evt BO Status/Reason Initial Load
D1-DETDX	Device Event Type Dimension Extract
D1-DETIL	Device Event Type Initial Load
D1-DEVFX	Device Event Accumulation Fact Extract
D1-DEVL	Device Event Initial Load
D1-DVCDX	Device Dimension Extract
D1-DVCIL	Device Dimension Initial Load
D1-IESDX	Install Evt BO Status/Reason Dim Extract
D1-IESIL	Install Evt BO Status/Reason Initial Load
D1-INEFX	Install Event Fact Extract
D1-INEIL	Install Event Initial Load
D1-LNMDX	Days Since Last Normal Measurement Extract
D1-LNML	Days Since Last Normal Msrmt Initial Load
D1-MCDX	Measuring Component Dimension Extract
D1-MCIL	Measuring Component Dimension Initial Load
D1-SPAFX	Service Point Accumulation Fact Extract
D1-SPDX	Service Point Dimension Extract
D1-SPIL	Service Point Initial Load
D1-SPRDX	Service Provider Dimension Extract
D1-SPRIL	Service Provider Initial Load
D1-SPSDX	SP BO Status/Reason Dimension Extract
D1-SPSFX	Service Point Snapshot Fact Extract

Batch Control	Name / Description
D1-SPSIL	SP BO Status/Reason Initial Load
D2-CONDX	Contact Dimension Extract
D2-CONIL	Contact Intial Load
D2-CSTDX	Usage Snapshot Type Dimension Extract
D2-CSTIL	Usage Snapshot Type Initial Load
D2-EXLIL	Exception Severity Lookup Extract
D2-EXTDX	Exception Type Dimension Extract
D2-EXTIL	Exception Type Initial Load
D2-ITLIL	IMD Type Lookup Extract
D2-LUTDX	Days Since Last UT Dimension Extract
D2-LUTIL	Days Since Last UT Initial Load
D2-MRCDX	Measurement Condition Dimension Extract
D2-MRCIL	Measurement Condition Initial Load
D2-MVREF	Materialized View Refresh
D2-SPCFX	Usage Snapshot Fact Extract
D2-SUAFX	Unreported Usage Analysis Fact Extract
D2-SVEFX	VEE Exception Snapshot Fact Extract
D2-UGDX	Usage Group Dimension Extract
D2-UGIL	Usage Group Initial Load
D2-USDX	Usage Subscription Dimension Extract
D2-USIL	Usage Subscription Initial Load
D2-UTADX	Unreported Usage Analysis Type Dim Extract
D2-UTAIL	SP UT Aging Snapshot Type Initial Load
D2-UTIL	UOM/TOU Dimension Extract
D2-UTSIL	UOM/TOU/SQI Dimension Extract
D2-VERDX	VEE Rule Dimension Extract
D2-VERIL	VEE Rule Dimension Initial Load

Refer to the *Oracle Utilities Meter Data Management Business Intelligence Data Mapping Guide* for more information for more information about setting up batch processing for Oracle Utilities Meter Data Management Business Intelligence.

Chapter 15

Integrating Oracle Utilities Meter Data Management with Other Systems

This chapter provides information related to integrating Oracle Utilities Meter Data Management with other applications, including:

- **Integrating with a Customer Information System**
- **Integrating with Oracle Utilities Operational Device Management**
- **Integrating with Oracle Utilities Business Intelligence**
- **Integrating with Oracle Utilities Customer Self Service**
- **Initial Measurement and Device Event XML Formats**
- **Usage and Event Import - Message Driven Bean Configuration**

Integrating with a Customer Information System

This section provides an overview of how Oracle Utilities Meter Data Management supports integrations with a customer information system.

In an integration between Oracle Utilities Meter Data Management and a customer information system such as Oracle Utilities Customer Care and Billing:

- Oracle Utilities Meter Data Management is typically the “system of record” for meter-related data, including meter records, meter configurations, validation, editing, and estimation (VEE) rules, bill determinant calculation rules, usage data, and calculated bill determinants.
- Oracle Utilities Customer Care and Billing (the customer information system) is typically the “system of record” for account related and service point-related data, including the rates and tariffs used to calculate bills for each account and customer.

Given this breakdown of data between the two systems, any integration between them must account for the passage of data between the two to ensure that each system can accurately perform its business functions. The integration between Oracle Utilities Meter Data Management and Oracle Utilities Customer Care and Billing is based on two core business processes:

- **Data Synchronization**
- **Processing Usage Transaction Requests**

Data Synchronization

In most integrations with Oracle Utilities Customer Care and Billing (or other CIS), Oracle Utilities Meter Data Management is not used as the system of record for account, customer, or service point-related data. Synchronizing this data between the two systems ensures that all account, customer, and service point-related data in Oracle Utilities Meter Data Management is correct and up to date before usage transaction calculations are performed. This synchronization process is supported through a set of business objects, master configurations, batch controls, and pre-configured XAI Inbound Services.

Types of Requests

Data synchronization is performed via synchronization requests sent from Oracle Utilities Customer Care and Billing via a middleware integration component. Oracle Utilities Meter Data Management supports three types of synchronization requests:

Initial Synchronization Requests

Initial synchronization requests are used when initially setting up Oracle Utilities Meter Data Management. They facilitate import of data that creates devices, device configurations, measuring components, service points, install events, contacts, and usage subscriptions in Oracle Utilities Meter Data Management based on corresponding data in Oracle Utilities Customer Care and Billing.

Ongoing Synchronization Requests

Ongoing synchronization requests are used when updating existing data in Oracle Utilities Meter Data Management based on changes in corresponding data in Oracle Utilities Customer Care and Billing. Ongoing synchronization requests can be used to update contacts, devices, device configurations, measuring components, install events, service points, and usage subscriptions.

Composite Synchronization Requests

Composite synchronization requests are requests that contain synchronization requests for multiple types of data within a single request. For example, a composite request could contain requests to update device, device configuration, measuring component, and install event data. This supports situations where multiple types of data must be updated based on a single change in Oracle Utilities Customer Care and Billing.

Base Package Synchronization Request Business Objects

The table below lists the base package synchronization request business objects used by Oracle Utilities Meter Data Management.

Business Object	Description
D1-CompositeSyncRequest	Composite Sync Request Used as a parent business object to define the behavior of a composite synchronization request that contains multiple synchronization requests (defined as child business objects). It is designed to accept a sync request whose information needs to be propagated to multiple maintenance objects. Its children business objects contain the elements for all the maintenance objects involved and then, within their processing, create a sync request for each maintenance object instance required by the inbound request. For example, composite synchronization requests that contains requests for device, device configuration, and install event could use a composite business object with the Device, Device Configuration, and Install Event synchronization request business objects as child business objects. When processed, this business object would create instances of the Device, Device Configuration, and Install Event synchronization request business objects.
D1-CompositeSyncRequestDC	Device Config Composite Sync Request An example composite synchronization request. This business object a child of the D1-CompositeSyncRequest business object.
D1-InitialSyncRequestContact	Contact Initial Sync Request Instances of this business object represent individual initial contact synchronization requests.
D1-InitialSyncRequestDC	Device Configuration Initial Sync Request Instances of this business object represent individual initial device configuration synchronization requests.
D1-InitialSyncRequestDevice	Device Initial Sync Request Instances of this business object represent individual initial device synchronization requests.
D1-InitialSyncRequestIE	Install Event Initial Sync Request Instances of this business object represent individual initial install event synchronization requests.

Business Object	Description
D1-InitialSyncRequestMC	Measuring Component Initial Sync Request Instances of this business object represent individual initial measuring components synchronization requests.
D1-InitialSyncRequestSP	Service Point Initial Sync Request Instances of this business object represent individual initial service point synchronization requests.
D1-OngoingSyncReqAckMsg	Ongoing Sync Request Acknowledgement Used to send a message to the sending system to acknowledge receipt of an ongoing synchronization request.
D1-OngoingSyncReqScalarMtrRead	Scalar Meter Read Ongoing Sync Request Instances of this business object represent individual ongoing scalar meter read synchronization requests.
D1-OngoingSyncRequestContact	Contact Ongoing Sync Request Instances of this business object represent individual ongoing contact synchronization requests.
D1-OngoingSyncRequestDC	Device Configuration Ongoing Sync Request Instances of this business object represent individual ongoing device configuration synchronization requests.
D1-OngoingSyncRequestDevice	Device Ongoing Sync Request Instances of this business object represent individual ongoing device synchronization requests.
D1-OngoingSyncRequestIE	Install Event Ongoing Sync Request Instances of this business object represent individual ongoing install event synchronization requests.
D1-OngoingSyncRequestMC	Measuring Component Ongoing Sync Request Instances of this business object represent individual ongoing measuring component synchronization requests.
D1-OngoingSyncRequestSP	Service Point Ongoing Sync Request Instances of this business object represent individual ongoing service point synchronization requests.
D1-SeederSyncMasterConfig	Seeder Sync Request Master Configuration Used to define the foreign keys for each entity being synchronized and their corresponding views that will be used in translating the external (old) key to production (new) key references during initial conversion processing or ongoing synchronization processing.

Business Object	Description
D1-SyncRequestSeeder	Sync Request Seeder Used to identify the appropriate synchronization business object to create when processing synchronization requests.
D1-SynchronizationAddContact	Contact Synchronization Add Used when adding a new contact as a result of a synchronization request.
D1-SynchronizationAddDC	Device Configuration Synchronization Add Used when adding a new device configuration as a result of a synchronization request.
D1-SynchronizationAddDevice	Device Synchronization Add Used when adding a new device as a result of a synchronization request.
D1-SynchronizationAddIE	Install Event Synchronization Add Used when adding a new install event as a result of a synchronization request.
D1-SynchronizationAddMC	Measuring Component Synchronization Add Used when adding a new measuring component as a result of a synchronization request.
D1-SynchronizationAddSP	Service Point Synchronization Add Used when adding a new service point as a result of a synchronization request.
D2-InitialSyncRequestUS	Usage Subscription Initial Sync Request Instances of this business object represent individual initial usage subscription synchronization requests.
D2-OngoingSyncRequestUS	Usage Subscription Ongoing Sync Request Instances of this business object represent individual ongoing usage subscription synchronization requests.
D2-SynchronizationAddUS	Usage Subscription Synchronization Add Used when adding a new usage subscription as a result of a synchronization request.

Master Configurations

Master configurations are used to define aspects of the synchronization process, including resolution of foreign keys and the type of synchronization business objects to use for each type of data being synchronized.

The table below lists the master configurations used in data synchronization processing.

Master Configuration	Description
Master Data Synchronization Configuration	Lists all foreign key references that need resolution. Each one should reference the view that contains the external key / production key cross-reference. For entities that undergo both the initial and the ongoing sync, two views are specified. For entities that undergo the ongoing sync, an external system / ID type mapping is specified to cater for entities that might be synchronizing from more than one external system.
Seeder Sync Request Master Configuration	Lists the maintenance objects (device, device configuration, etc.) that require synchronization. Each references the synchronization business object that needs to be instantiated when processing a synchronization request for that maintenance object. For maintenance objects that undergo both initial and the ongoing synchronization, two business objects are specified.

Refer to the Oracle Utilities Application Framework documentation for more information about master configurations.

Batch Controls

Batch controls perform processing for initial synchronization requests such as allocating keys to data, resolving foreign keys, and loading data (instantiating business objects representing entities such devices, measuring components, etc.).

“Initial Sync Request - Resolve Keys XXX” batch controls invoke a generic maintenance object transition process to invoke the “Resolve Keys - Initial Sync” algorithm for synchronization requests of the appropriate type. Parameters used by “resolve keys” batch controls include:

- **Maintenance Object**: (Required) the maintenance object (device, device configuration, etc.) to be processed. This must be set to the Sync Request maintenance object for the batch control (device for device synchronization requests, service point for service point synchronization requests, etc.)
- **Restrict By Batch Code**: Restricts processing to synchronization requests whose current state is linked to this batch code.
- **Restrict By Business Object**: Restricts processing to synchronization requests linked to this business object.
- **Restrict By Status Code**: Restricts processing to synchronization requests of this status (default: KEY_ALLOCATD).
- **Max Errors**: Specifies the maximum number of errors allowed before the process exits.

"Initial Sync Request - Load Data XXX" batch controls load data (created new instances of business objects) for requests of the appropriate type (device, measuring component, etc,). Parameters used by "load data" batch controls include:

- **Maintenance Object**: (Required) the maintenance object (device, device configuration, etc.) to be processed. This must be set to the Sync Request maintenance object for the batch control (device for device synchronization requests, service point for service point synchronization requests, etc.)
- **Restrict By Batch Code**: Restricts processing to synchronization requests whose current state is linked to this batch code.
- **Restrict By Business Object**: Restricts processing to synchronization requests linked to this business object.
- **Max Errors**: Specifies the maximum number of errors allowed before the process exits.

The table below lists the batch controls used by initial synchronization requests

Batch Control	Description
D1-CMSYN	Composite Sync Request Transition Composite Sync Request BO instances in Pending state to the next state.
D1-SIIER	Initial Sync Request - Error Transition Initial Inbound Sync Request BO instances in Error state to the next state.
D1-SILCN	Initial Sync Request - Load Data Contact
D1-SILDC	Initial Sync Request - Load Data DC
D1-SILDV	Initial Sync Request - Load Data Device
D1-SILIE	Initial Sync Request - Load Data IE
D1-SILMC	Initial Sync Request - Load Data MC
D1-SILSP	Initial Sync Request - Load Data SP
D1-SILUS	Initial Sync Request - Load Data US
D1-SIKCN	Initial Sync Request - Resolve Keys Contact
D1-SIKDC	Initial Sync Request - Resolve Keys DC
D1-SIKDC	Initial Sync Request - Resolve Keys Device
D1-SIKIE	Initial Sync Request - Resolve Keys IE
D1-SIKMC	Initial Sync Request - Resolve Keys MC
D1-SIKSP	Initial Sync Request - Resolve Keys SP
D1-SIKUS	Initial Sync Request - Resolve Keys US
D1-SIOER	Ongoing Sync Request -Error Transition Ongoing Inbound Sync Request BO instances in Error state to the next state.
D1-SIOPE	Ongoing Sync Request - Pending Transition Ongoing Inbound Sync Request BO instances in Pending state to the next state.

Batch Control	Description
D1-SRSDE	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
F1-SAKRQ	Sync Request Allocate Keys Monitor Allocates keys to the entities that needs to be added. Batch parameters govern whether the processing is further restricted by batch code, business object, maintenance object and max errors.
F1-SRLRQ	Sync Request Load Records Monitor Performs fast inserts using batch add.
F1-SYNRQ	Sync Request Monitor Process Invokes monitoring rules associated with the current state of sync requests. All monitoring rules throughout the sync request's business object's inheritance chain are considered. Batch parameters govern whether the processing is further restricted by batch code, business object and status.
F1-SYSRQ	Sync Request Sampling Monitor (Deferred) Invokes monitoring rules associated with the current state of inbound synchronization requests. All monitoring rules throughout the inbound synchronization request's business object's inheritance chain are considered.

Batch Control Scheduling

The following table specifies the order in which the batch controls on the Initial Sync Request BO life cycle should be executed. The first row identifies the maintenance object for which the synchronization request is intended and the first column specifies the type of process.

	Contact	US	SP	Install Event	Device Config	Device	Measuring Component
Transformation / Schema Validation	1	1	1	1	1	1	1
Allocate Keys	2	2	2	2	2	2	2
Foreign Key Resolution /BO Validation	3	8	3	6	3	3	3
Load Data	4	9	5	7	4	4	4

Note that before the Key Resolution job is run, all the Key Allocation Jobs need to finish. This ensures that all foreign key references can be subsequently resolved.

Some business object-level validation is dependent on other entities being completely loaded first. The sequence numbers above allow for this. For example, usage subscriptions business object validation is dependent on service points existing; Install Event business object validation is dependent on both service points and devices existing.

XAI Inbound Services

XAI inbound services are used to facilitate invoking the Sync Request Seeder business object by the middleware components upon receipt of a synchronization request.

The table below lists the pre-configured XAI Inbound Services used to process synchronization requests sent from Oracle Utilities Customer Care and Billing.

XAI Inbound Service	Description	Schema Name
D1-SyncRequestInbound	Sync Request Inbound	D1-SyncRequestSeeder (BO)
D1-SyncRequestInboundComposite	Sync Request Inbound Composite	D1-CompositeSyncRequestDC (BO)

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

Understanding Synchronization Request Processing

This section provides an overview of the processing that takes place when a synchronization request is sent. For each step in the process, the table below provides a brief description of the processing that takes place, and lists the specific objects involved.

Step	Process	Objects
1.	Oracle Utilities Customer Care and Billing sends a synchronization request to the middleware integration layer. For example, consider a request to update information about a service point.	
2.	The middleware components transforms the request from the Customer Care and Billing format, to the format used by Oracle Utilities Meter Data Management (this format is based on the business object schemas of the synchronization request business objects).	
3.	The middleware component invoke the appropriate XAI Inbound Service, and sends the transformed request.	XAI Inbound Service: D1-SyncRequestInbound (mapped to the D1-SynrRequestSeeder business object)
4.	The XAI Inbound Service invokes the Sync Request Seeder business object, which in turn, determines which synchronization request business object to create (based on the type of data in the synchronization request and the Seeder Sync Master Configuration).	Synchronization Request BO: D1-OngoingSyncRequestSP
5.	For initial synchronization requests, background processing creates master data for each synchronization request, including the following steps: • Data Transformation / Schema Validation • Allocate Keys • Resolve Foreign Keys / Validate Business Object • Load Data See **Batch Control Scheduling** on page 15-8 for more information about scheduling batch processing for initial synchronization requests.	

Step	Process	Objects

6.	For ongoing synchronization requests, the following steps are performed by Enter algorithms on the synchronization request business object lifecycle states:
	• Data Transformed/Basic Schema Validated:
	  • Determine Sync Request BO
	  • Setup Transformed Data
	• Pre-Added:
	  • Sync Request Pre-Add Data
	• FKs Resolved:
	  • Resolve Keys - Ongoing Sync
	• Updating:
	  • Sync Request Update Data
		Setup Transformed Data Algorithm: D1-SETTRANDT
		
		Sync Request Pre-Add Data Algorithm: D1-SR-PREADD
		
		Resolve Keys - Ongoing Sync Algorithm: D1-RESKEYFAL
		
		Sync Request Update Data Algorithm: D1-SR-UPDDAT

Processing Usage Transaction Requests

Oracle Utilities Customer Care and Billing uses bill determinant data (usage transactions based on meter readings) calculated and stored in Oracle Utilities Meter Data Management when calculating bills for customers. This process allows Oracle Utilities Customer Care and Billing to send requests for usage transaction calculations to Oracle Utilities Meter Data Management, which in turn performs the requested calculations, and publishes the results back to Oracle Utilities Customer Care and Billing. Processing usage transaction requests is supported through a set business objects, pre-configured XAI Inbound services, and processing methods.

Business Objects

The table below lists the business objects used when processing usage transaction requests.

Business Object	Description
D2-UsgTranSeeder	Usage Transaction Seeder Used to determine the usage transaction business object to use when creating new usage transactions.

XAI Inbound Services

XAI inbound services are used to facilitate invoking the Usage Transaction Seeder business object by the middleware components upon receipt of a usage transaction request.

The table below lists the pre-configured XAI Inbound Services used to process usage transaction requests sent from Oracle Utilities Customer Care and Billing.

XAI Inbound Service	Description	Schema Name
D2-UsageTransactionRequestInbound	Usage Transaction Request Inbound	D2-UsgTranSeeder (BO)

Processing Methods

Processing methods are used to determine the usage transaction business object to use when creating usage transactions based on the requests, and for determining the method by which usage transactions are sent back to Oracle Utilities Customer Care and Billing.

Processing Method	Description
D2-HowToSendUSInfoOnline	How To Send US Information - Online Used to specify the method (batch control, business object, or outbound message) by which usage transactions are sent to subscribing systems in real time.
D2-HowToSendUSInformation	How To Send US Related Information Used to specify the method (batch control, business object, or outbound message) by which usage transactions are sent to subscribing systems.
D2-HowToCreateUSInformation	How To Create US Related Information Used to determine the type of usage transaction business object to create.
D2-HowToSendUSInfoBatch	How To Send US Related Information Batch Used to specify the method (batch control, business object, or outbound message) by which usage transactions are sent to subscribing systems via batch processing.

Understanding Usage Transaction Request Processing

This section provides an overview of the processing that takes place when a usage transaction request is sent. For each step in the process, the table below provides a brief description of the processing that takes place, and lists the specific objects involved.

Step	Process	Objects
1.	Oracle Utilities Customer Care and Billing sends a usage transaction request to the middleware integration layer.	
2.	The middleware components transforms the request from the Customer Care and Billing format, to the format used by Oracle Utilities Meter Data Management (this format is based on the business object schemas of the synchronization request business objects).	
3.	The middleware component invoke the appropriate XAI Inbound Service, and sends the transformed request.	XAI Inbound Service: D2-UsageTransactionRequestInbound (mapped to the D2-UsgTranSeeder business object)

Step	Process	Objects
4.	The XAI Inbound Service invokes the Usage Transaction Seeder business object. This business object: • Determines the usage subscription ID based on an external usage subscription ID This processing is performed via a Pre-Processing algorithm. • Determines the appropriate usage transaction business object to create. This processing is performed via a Pre-Processing algorithm and a processing method defined for the “Usage Transaction Creation” processing role on the service provider that represents the external system.	Pre-Processing Algorithm: D2-DETUSID (Determine Usage Subscription ID) Pre-Processing Algorithm: D2-DETUTBO (Determine Usage Transactions Business Object) Processing Method: D2-HowToCreateUSInformation (How To Create Usage Subscription Related Information)
5.	The usage transaction determines the usage group(s) to use when calculating usage. These are retrieved from the usage subscription (with a fallback to the usage subscription type). This base package logic can be overridden by specifying an algorithm in the **Usage Group Determination Override Algorithm** field on the usage subscription type. If an algorithm is specified in this field, its logic overrides the existing method of determining the usage group. A base package algorithm (D2-DRVUSGGRP) is provided that can be used here. This algorithm uses the **CCB Rate Schedule** extendable lookup used to map the rate schedules in Oracle Utilities Customer Care and Billing to usage groups. The algorithm's logic looks for a combination of rate history list entries on the usage subscription and the type of device configuration installed during the bill period to determine the usage group(s) to use for bill determinants calculation.	
5.	The usage transaction business object calculates usage based on the date range provided in the request. This processing is performed by an Enter algorithm on the “Calculate” state of the usage transaction business object.	Enter Algorithm: D2-CALCUSG (Calculate Usage)
6.	If the usage transaction has a sub usage transactions, it checks the status of each. This processing is performed by an Enter algorithm on the “Calculation in Progress” state of the usage transaction business object.	Enter Algorithm: D2-CHKSUBUT (Check Sub Usage Transactions)

Step	Process	Objects
7.	If the usage transaction is configured to require an approval, a To Do Entry is created. This processing is performed by an Enter algorithm on the "Approval in Progress" state of the usage transaction business object.	Enter Algorithm: D2-UTAP-TODO (Create Usage Transaction Requiring Approval To Do Entry)
8.	The usage transaction then sends the usage transaction to any subscribing systems. This processing is performed by an Enter algorithm on the "Sent" state of the usage transaction business object, and a processing method for the service provider for the subscribing systems.	Enter Algorithm: D2-SEND-USG (Send Usage) Processing Method: D2-HowToSendUSInformation (How To Send US Related Information)

Integrating with Oracle Utilities Operational Device Management

This section provides an overview of how Oracle Utilities Meter Data Management supports integrations with Oracle Utilities Operational Device Management.

In an integration between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management:

- Oracle Utilities Meter Data Management is typically considered the "system of record" for service points (asset locations) and contacts, and for device installation information (MDM Install Events)
- Oracle Utilities Operational Device Management is typically considered the "system of record" for assets (devices)

Given this breakdown of data between the two systems, any integration between them must account for the passage of data between the two to ensure that each system can accurately perform its business functions. The integration between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management is based on data synchronization between the two systems.

Data Synchronization

The specific data synchronization flows supported between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management include the following:

- **Asset-Device Synchronization**: As new assets are created or changed in Oracle Utilities Operational Device Management, corresponding devices must be created or changed in Oracle Utilities Meter Data Management
- **Service Point/Contact – Asset Location/Contact**: As Service Points and/or Contacts are created or changed in Oracle Utilities Meter Data Management, corresponding Asset Locations and Contacts must be created or changed in Oracle Utilities Operational Device Management
- **Install Events– Asset Location/Disposition**: As devices are installed/removed in Oracle Utilities Meter Data Management, corresponding changes to an asset's Disposition (location and status) must be made in Oracle Utilities Operational Device Management

This synchronization process is supported through a set of business objects, master configurations, batch controls, and pre-configured XAI Inbound Services. Refer to the *Oracle Utilities Integration for Device Operations Implementation Guide* for more information about this integration and the data synchronization processes used by this integration.

Inbound Data Synchronization Business Objects

The integration between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management uses the following inbound (to Oracle Utilities Mater Data Management) synchronization business objects (based on the F1-SYNCREQIN maintenance object):

Business Object	Description
D1-OngoingSyncRequestDevice	Device Ongoing Sync Request Used to synchronize devices in Oracle Utilities Mater Data Management based on assets created/updates in Oracle Utilities Operational Device Management

Outbound Data Synchronization Business Objects

The integration between Oracle Utilities Meter Data Management and Oracle Utilities Operational Device Management uses the following outbound (from Oracle Utilities Mater Data Management) synchronization business objects (based on the F1-SYNC REQ maintenance object)

Business Object	Description
D1-ODMContactSyncRequest	ODM Sync Request Used to synchronize contacts in Oracle Utilities Operational Device Management based on contacts created/updated in Oracle Utilities Meter Data Management
D1-ODMInstallEventSyncRequest	ODM Install Event Sync Request Used to synchronize asset location/disposition in Oracle Utilities Operational Device Management based on install events created/updated in Oracle Utilities Meter Data Management
D1-ODMSPSyncRequest	OMD SP Sync Request Used to synchronize asset locations in Oracle Utilities Operational Device Management based on service points created/updated in Oracle Utilities Meter Data Management

Master Configurations

Master configurations are used to define aspects of the synchronization process, including resolution of foreign keys and the type of synchronization business objects to use for each type of data being synchronized.

The table below lists the master configurations used in data synchronization processing.

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

Master Configuration	Description
Seeder Sync Request Master Configuration	Lists the maintenance objects (device, device configuration, etc.) that require synchronization. Each references the synchronization business object that needs to be instantiated when processing a synchronization request for that maintenance object. For maintenance objects that undergo both initial and the ongoing synchronization, two business objects are specified.
ODM Integration Master Configuration	Specifies the external system used to represent Oracle Utilities Operational Management, and the URL for the Oracle Utilities Operational Management application. Also specifies the outbound message types used to send synchronization requests to Oracle Utilities Operational Management.

Integrating with Oracle Utilities Business Intelligence

Oracle Utilities Meter Data Management can also be integrated with Oracle Utilities Business Intelligence to allow uses to view analytic data based on usage, events, and other data tracked in Oracle Utilities Meter Data Management.

Oracle Utilities Meter Data Management Business Intelligence Products

Oracle Utilities Meter Data Management Business Intelligence comprises the following products:

- **Oracle Utilities Meter Data Analytics**: dashboards, dashboard pages, and analytics used to view usage, event, and other data from Oracle Utilities Meter Data Management
- **Oracle Utilities Meter Data Schema and Extracts**: the star schema and extract programs used by Oracle Utilities Meter Data Analytics.

Business Intelligence Batch Controls

Oracle Utilities Meter Data Management data extract and initial load is performed via a set of batch controls provided in the base package. See **Meter Data Management Business Intelligence Batch Controls** on page 14-9 for a list of batch controls used for extract and initial load of data for use with Oracle Utilities Business Intelligence and Oracle Utilities Meter Data Analytics.

For More Information

Refer to the following documentation for more information about Oracle Utilities Meter Data Management Business Intelligence:

- Oracle Utilities Meter Data Management Business Intelligence Data Mapping Guide
- Oracle Utilities Meter Data Management Business Intelligence Metric Reference Guide
- Oracle Utilities Advanced Spatial and Operational Analytics User’s Guide

Integrating with Oracle Utilities Customer Self Service

Oracle Utilities Meter Data Management can be integrated with Oracle Utilities Customer Self Service allow utilities to allow their customers to view their usage data and create self-service meter readings.

Oracle Utilities Customer Self Service provides the following services based on Oracle Utilities Meter Data Management:

- **Create Self-Service Meter Read**: Allows users to submit their own meter reads via the Oracle Utilities Self Service application
  - **Service Script**: WX-CrSSMRead
  - **XAI Inbound Service**: WX-CreateSelfServiceMeterRead
- **Get Scalar Consumption Summary**: Retrieves consumption data for display in the Oracle Utilities Self Service application
  - **Service Script**: WX-GetSCsum
  - **XAI Inbound Service**: WX-GetScalarConsumptionSummary
- **Get Usage Overview**: Retrieves an overview of a customer's usage for a user-specified duration for display in the Oracle Utilities Self Service application
  - **Service Script**: WX-GetUsgOVw
  - **XAI Inbound Service**: WX-GetUsageOverview
- **Get Usage Details**: Retrieves usage details for a customer for a user-specific time period (year, month, day) for display in the Oracle Utilities Self Service application
  - **Business Service**: WX-RETWSSTOUMapping
  - **XAI Inbound Service**: WX-RETWSSTOUMappingService

These services are based on the service scripts and business services noted above, and are invoked via the corresponding XAI Inbound Services.

Refer to the *Oracle Utilities Customer Self Service Implementation Guide* for more information about integrating Oracle Utilities Meter Data Management with Oracle Utilities Customer Self Service.

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


```xml
<IMD-IMPORT>
  <serviceProvider>HEADEND-1</serviceProvider>
  <serviceProviderExternalId>MDCS-1</serviceProviderExternalId>
  <preVEE>
    <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>
    <inShift>N</inShift>
    <mcm>1.0</mcm>
    </nd>
    <tz>USPACIFIC</tz>
    <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>
</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>
    <inShift>N</inShift>
    <mcm>1.0</mcm>
    <nd>5</nd>
    <tz>USPACIFIC</tz>
    </ccond>
    <sts>
```

```
      <stsL>
        <s>1</s>
        <st>REGULAR</st>
      </stsL>
    </sts>
  </preVEE>
</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
<dvcIdN>	Device Identifier Number, e.g. a Serial Number. The "type" of device identifier that the head-end system understands is configured within MDM - and so can differ per head-end.
<mcId>	Measuring Component ID
<mcIdN>	Measuring Component Identifier Number. Populated with channel ID. The "type" of measuring component identifier that the head-end system understands is configured within MDM
<externalId>	File name of ID of the XML document containing the measurement data
<uom>	Unit of Measure (Optional)
<stDt>	Start Date/Time. Required for interval measurement data. Optional for scalar measurement data. Must be in the following format: YYYY-MM-DD-HH.MM.SS Example:2008-12-31-00.30.00
<stQty>	Start Reading. Optional.
<enDt>	End Date/Time. Required. Must be in the following format: YYYY-MM-DD-HH.MM.SS Example:2008-12-31-00.30.00

Element	Description
<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.
<inShift>	Incoming Data Shift flag. Indicates if the device is DST aware.
<mcm>	Meter multiplier.
<nd>	Number of Dials
<tz>	Time Zone
<spi>	Seconds-per-interval
<ccond>	Measurement condition
<sts>	Element containing status code information for entire measurement.
<stsL>	List of head-end system status codes for the entire measurement
<s>	Sequence
<st>	Head-end status codes applicable to the entire set of data.
<msrs>	Element containing measurement data
<mL>	Element containing an individual interval measurements. Used for interval measurement data only.
<s>	Sequence of the interval measurement
<dt>	Date and time of the interval measurement. Optional.
<q>	Quantity of the interval measurement
<ue>	Used-Edited flag. Indicates if the interval measurement has been manually edited
<fc>	Final Condition code for the interval measurement. Optional.
<sts>	Element containing lists of status codes for each interval measurement

Element	Description
<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.

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="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: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>
    <externalStatusValue>Tamper indication on serial number GD_LL_SN100.</
externalStatusValue>
    <externalStatusDateTime>2010-09-09-13.11.41</externalStatusDateTime>
  </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.

Element	Description
<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.
<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.
<externalStatusValue>	Optional information related to the event.
<externalStatusDateTime>	Date and time at which optional information (specified in the <externalStatusValue> element was recorded.

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="externalCommandId" type="xsd:string"
minOccurs="0" />
            </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 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**

JMS Configuration

JMS configuruation 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 **ejb-weblogic-jar.xml** configuration files delivered with Oracle Utilities Smart Grid Gateway. It is recommended that instead of modifying these files directly you create "Customer Modification" (CM) versions of these files to make changes to these configuration files. This ensures that your modifications are not overwritten by future application patches.

The following section describes the changes required in the CM files for configuring the MDBs to read from the foreign JMS queues set up in the steps above (see **JMS Configuration** on page 15-33). 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.

Chapter 16

Sample Implementation

This chapter describes the steps involved in configuring Oracle Utilities Meter Data Management in a simple example implementation, including the following:

- **Implementation Description and Requirements**
- **Implementation Steps**

Note: The implementation described in this chapter is intended for example purposes only, and is intentionally simple, and as such does not involve configuration of every type of object described in this book. Also, this example assumes that the base package admin business objects (device type, measuring component type, etc. meet the requirements of the implementation.

Implementation Description and Requirements

The sample implementation described in this chapter 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 • Commercial: Interval channel/scalar register • Industrial: Two interval channels
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)
Validation, Editing, and Estimation Rules	All meters: • Meter Multiplier Check • UOM Check • Device ID Check Interval meters: • Interval spike check • Interval size validation • Interval replacement • Sum Check (commercial only) • KVARH Check (industrial only) - identifies intervals where reactive load (kVARh) is present and active load (kWh) is not. Scalar meters: • Scalar Replacement • Negative consumption
Bill Determinants	• Residential: KWH • Commercial: KWH and KW (demand) • Industrial: On-Peak KWH, Off-Peak KWH, and KVA (demand)
Billing System	Oracle Utilities Customer Care and Billing

Implementation Steps

This section outlines the steps in configuring Oracle Utilities Meter Data Management to meet the requirements outlined above. These steps include:

1. **Design and Create Business Objects**

   In this first step, we'll outline the specific business objects and other configuration data required to address the sample requirements.

2. **Create Admin Data**

   In this step, we'll outline the admin data that would need to be created to address the sample requirements.

3. **Create Master Data**

   In this step, we'll outline the master data (individual devices, service points, etc.) that would need to be created to address the sample requirements.

Design and Create Business Objects

The first step in implementing and configuring the system is to identify the business objects (and related configuration objects) needed to meet the requirements of the implementation. This section outlines the business objects and other significant configuration objects that could be created to meet the requirements of our sample implementation. This does NOT include listings of all configuration objects needed (such as individual display and maintenance UI maps, portal navigation options, etc.).

Service Points and Device Installation

For service point and device installation data, we would need to create the following:

Service Points: Service point business objects for each type of customer, as follows:

- Residential: CM-ResidentialSP
- Commercial: CM-CommercialSP
- Industrial: CM-IndustrialSP

Contacts: Contact business objects for each type of customer, as follows:

- Residential: CM-ResPerson
- Commercial: CM-ComBusiness
- Industrial: CM-IndBusiness

Install Events: Install event business objects for each type of customer, as follows:

- Residential: CM-ResidentialMeterInstallEvent
- Commercial: CM-CommercialMeterInstallEvent
- Industrial: CM-IndustrialMeterInstallEvent

Service Providers: Service provider business objects for each head-end system, and for the billing system, as follows:

- Head-End System: CM-HeadEndMRU (Meters R Us)
- Head-End System: CM-HeadEndMT (Meter Tech, Inc.)
- Billing System: CM-ExternalAppCCB

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

Devices and Measuring Components

For devices and measuring components, we would need to create the following:

Devices: Device business objects for each type of meter, as follows:

- Residential: CM-ScalarRegister
  - Install Event BO: CM-ResidentialInstallEvent (see above)
- Commercial: CM-IntChanScalarReg
  - Install Event BO: CM-CommercialInstallEvent (see above)
- Industrial: CM-Interval2Channels
  - Install Event BO: CM-IndustrialInstallEvent (see above)

Measuring Components: Measuring component business objects for scalar registers and/or interval channels, as follows:

- Residential - Scalar Register: CM-ResScalarRegister
- Commercial/Industrial - Interval Channel: CM-IntervalChannel (used for both commercial and industrial meters)
- Commercial - Scalar Register: CM-ScalarValRegister

Measurement Data

For measurement data, we would need to create the following:

Initial Measurement Data: Initial load, estimation, and manual initial measurement business objects for each reading/measurement type, as follows:

- Initial Load - Interval: CM-InitialLoadIMDInterval
- Initial Load - Scalar: CM-InitialLoadIMDScalar
- Estimation - Interval: CM-EstimationIMDInterval
- Estimation - Scalar: CM-EstimationIMDScalar
- Manual - Interval: CM-ManualIMDInterval
- Manual - Scalar: CM-ManualIMDScalar

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

- Final Measurement: CM-FinalMeasurement

VEE Groups and Rules

For VEE groups and rules, we would need to the create the following:

VEE Rules: Business object and algorithm type / algorithm for the KVARH Check validation, as follows:

- Business Object: CM-KVARHCheck
- Algorithm Type: CM-KVARHCHK
- Algorithm: CM-KVARHCHK

Usage Subscriptions

For usage subscriptions, we would need to create the following:

Usage Subscriptions: Usage subscription business object for each type of customer, as follows:

- Residential: CM-ResidentialUS
- Commercial: CM-CommercialUS
- Industrial: CM-IndustrialUS

Usage Groups and Rules

For usage groups and rules, we would need to the create the following:

Usage Rules: Business object and algorithm type / algorithm for the KVA calculation, as follows:

- Business Object: CM-CalculateKva
- Algorithm Type: CM-CALCKVA
- Algorithm: CM-CALCKVA

TOU Maps and Dynamic Options

For TOU maps and dynamic options, this implementation can use the base package business objects.

Usage Transactions

For usage transactions, we would need to create the following:

Usage Transactions: Usage transaction business object for each type of customer, as follows:

- Residential: CM-ResidentialUT
- Commercial: CM-CommercialUT
- Industrial: CM-IndustrialUT

Create Admin Data

With all of the custom business objects needed for the implementation in place, the next step would be to create admin data. This section outlines the admin data that would need to be created to meet the requirements of our sample implementation. In general these listings list only the name (or code) and description of each record to be created, and do not include details for every attribute of each record created. Where listing additional attributes is important to understanding how the data would be created, it is noted, and additional details are provided in a separate section.

The table below summarizes the common admin data needed for our implementation

Admin Data Type	Data to Create
Activity Type	N/A (will use base package activity types)
Contact Type	One for each type of customer rule, as follows: • RESIDENTIAL (Residential - Person) Business Object: CM-ResPerson • COMMERCIAL (Commercial - Business) Business Object: CM-ComBusiness • INDUSTRIAL (Industrial - Business) Business Object: CM-IndBusiness
Exception Type	One for each VEE rule, as follows: • MMULTCHK (Meter Multiplier Check) • UOMCHK (UOM Check) • DVCICCHK (Device ID Check) • INTSPIKECHK (Interval spike check) • INTSIZEVAL (Interval size validation) • INTREPL (Interval replacement) • INTKVARHCHK (Interval KVARH Check) • SCALREPL (Scalar Replacement) • NEGCONS (Negative consumption)
Factor	N/A
Market	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
	• Etc.

Service Provider	One for each head-end system, and one for the billing system, as follows: • METERSRUS (Meters R Us) Business Object: CM-HeadEndMRU • METERTECH (MeterTech, Inc.) Business Object: CM-HeadEndMT • OUCCB (Oracle Utilities Customer Care and Billing) Business Object: CM-ExternalAppCCB * See **Service Providers** on page 16-11 for additional details.
Service Quantity Identifier	N/A
Service Type	Single service type: • ELECTRIC (Electric)
Time of Use	One for each time of use period, as follows: • ONPEAK (On Peak) • OFFPEAK (Off Peak)
VEE Group	See **VEE Groups and Rules** on page 16-12 for details.
Dynamic Option Type	N/A

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

TOU Group	One group for TOU periods, as follows: • ON-OFF_TOU_GRP (On-Peak / Off Peak TOU Group) - Contains ONPEAK and OFFPEAK TOU periods.
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.

VEE Rule	See **VEE Groups and Rules** on page 16-12 for details.
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 16-13 for additional details.
TOU Map Template	One TOU map template for each TOU schedule, as follows: • IND_TOU_SCHED_1 (TOU Schedule 1) • IND_TOU_SCHED_2 (TOU Schedule 2) • Etc. * See **TOU Map Templates and TOU Map Types** on page 16-15 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.
TOU Map Type	One TOU map type for each of three interval sizes, as follows: • IND_TOU_MAP_15_MIN (TOU Map Type - 15 Minutes) • IND_TOU_MAP_30_MIN (TOU Map Type - 30 Minutes) • IND_TOU_MAP_60_MIN (TOU Map Type - 60 Minutes) * See **TOU Map Templates and TOU Map Types** on page 16-15 for additional details.
Usage Group	One usage group for each type of customer, as follows: • RES_USAGE_GRP (Residential Usage Rules) Valid Device Configuration Types: • RESDVCCFG (Residential Device Configurations) • COM_USAGE_GRP (Commercial Usage Rules) Valid Device Configuration Types: • COMDVCCFG (Commercial Device Configurations) • IND_USAGE_GRP (Residential Usage Rules) Valid Device Configuration Types: • INDDVCCFG (Industrial Device Configurations) * See **Usage Groups and Rules** on page 16-16 for additional details.

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
Usage Rule	Five usage rules, as follows: • RES_SCALAR_DETAILS (Retrieve KWH for Scalar Register) • COM_APPLY_MATH_KWH (Calculate KWH for Interval Channel) • COM_APPLY_MATH_KW (Calculate KW for Interval Channel) • IND_GET_TOU_DATA (Calculate TOU-based KWH for interval channel) • CALC_KVA (Calculate KVA from KWH and KVARH interval channels) Business Object: CM-CalculateKva * See **Usage Groups and Rules** on page 16-16 for additional details.
Usage Subscription Type	One for each type of customer rule, as follows: • RESIDENTIAL (Residential) Business Object: CM-ResidentialUS Usage Recipient: OUCCB (Oracle Utilities Customer Care and Billing) • COMMERCIAL (Commercial) Business Object: CM-CommercialUS Usage Recipient: OUCCB (Oracle Utilities Customer Care and Billing) • INDUSTRIAL (Industrial) Business Object: CM-IndustrialUS Usage Recipient: OUCCB (Oracle Utilities Customer Care and Billing) * See **Usage Subscription Types** on page 16-16 for additional details.

Additional Details

This section provides additional details related to the admin data described above. Not all attributes are listed for all types of data.

Service Providers

This section provides additional details for each of the service providers listed above.

Service Provider: METERSRUS

- Business Object: CM-HeadEndMRU
- Description: Meters R Us
- External Reference ID: HE-MRU
- Our Name/ID in Their System: HE-MRU-11
- Processing Methods List:
  - Initial Measurement Creation (How To Create MC Related Information)
    - Default Processing Method (Business Object): CM-InitialLoadIMDScalar
    - Override Process Method:

Measuring Component Type	Business Object
Residential Scalar Register (RESSCALAR)	CM-InitialLoadIMDScalar

Service Provider: METERTECH

- Business Object: CM-HeadEndMT
- Description: MeterTech, Inc.
- External Reference ID: HE-MTECH
- Our Name/ID in Their System: HE-MTECH-11
- Processing Methods List:
  - Initial Measurement Creation (How To Create MC Related Information)
    - Default Processing Method (Business Object): CM-InitialLoadIMDInterval
    - Override Process Method:

Measuring Component Type	Business Object
Commercial Interval Channel (COMINTERVAL)	CM-InitialLoadIMDInterval
Commercial Scalar Register (COMSCALAR)	CM-InitialLoadIMDScalar
Industrial KVARH Interval Channel (INDINTERVALKVARH)	CM-InitialLoadIMDInterval
Industrial KWH Interval Channel (INDINTERVALKWH)	CM-InitialLoadIMDInterval

Service Provider: OUCCB

- Business Object: CM-ExternalAppCCB
- Description: Oracle Utilities Customer Care and Billing
- External Reference ID: EXT-CCB
- Our Name/ID in Their System: EXT-CCB-11
- Processing Methods List:
  - Usage Transaction Creation (How To Create Usage Subscription Related Information)
    - Default Processing Method (Business Object): CM-CommercialUT
    - Override Process Method:

Usage Subscription Type	Business Object
Residential (RESIDENTIAL)	CM-ResidentialUT
Commercial (COMMERCIAL)	CM-CommercialUT
Industrial (INDUSTRIAL)	CM-IndustrialUT

VEE Groups and Rules

The table below lists the VEE groups and corresponding VEE rules for this implementation.

VEE Group	VEE Rules
ALL_RULES Contains rules for all measuring components (referenced by VEE rule).	• MMULTCHK (Meter Multiplier Check) • UOMCHK (UOM Check) • DVCICCHK (Device ID Check)
INTERVAL_RULES Contains rules for interval measuring components (referenced by VEE rule).	• INTSPIKECHK (Interval Spike) • INTSIZEVAL (Interval Size Val) • INTREPL (Interval Replacement)
SCALAR_RULES Contains rules for scalar measuring components (referenced by VEE rule).	• SCALREPL (Scalar Replacement) • NEGCONS (Negative Consumption)
SCALAR_MCS Contain rules to be applied to scalar measuring components (including referenced groups).	• ALL_REF (References ALL_RULES) • SCALAR_REF (References SCALAR_RULES) • SCALAR_EXCEPTIONS (Exception Handler for Scalar)
COM_INTD_MCS Contain rules to be applied to commercial interval measuring components (including referenced groups).	• ALL_REF (References ALL_RULES) • INTERVAL_REF (References INTERVAL_RULES) • SUMCHK (Sum Check) • INTD_EXCEPTIONS (Exception Handler for Interval)

VEE Group	VEE Rules
IND_INTD_MCS Contain rules to be applied to industrial interval measuring components (including referenced groups).	• ALL_REF (References ALL_RULES) • INTERVAL_REF (References INTERVAL_RULES) • INTKVARHCHK (Interval KVARH Check) • INTD_EXCEPTIONS (Exception Handler for Interval)

Measuring Component Types

This section provides additional details for each of the measuring component types listed above.

Measuring Component Type: RESSCALAR

- Description: Residential Scalar Register
- Measuring Component Business Object: CM-ResScalarRegister
- Measurement Business Object: CM-FinalMeasurement
- Service Type: Electric
- Value Identifiers:

Value Identifier Type	Short-Hand Description	UOM
Measurement	KWH	KWH

- Valid VEE Groups:
    - Initial Load VEE Group - Scalar (SCALAR_MCS)
- Fallback VEE Groups:
    - Initial Load: Initial Load VEE Group - Scalar (SCALAR_MCS)

Measuring Component Type: COMINTERVAL

- Description: Commercial Interval Channel
- Measuring Component Business Object: CM-IntervalChannel
- Measurement Business Object: CM-FinalMeasurement
- Service Type: Electric
- Value Identifiers:

Value Identifier Type	Short-Hand Description	UOM
Measurement	KWH	KWH

- Valid VEE Groups:
    - Initial Load VEE Group - Commercial Interval (COM_INTD_MCS)
- Fallback VEE Groups:
    - Initial Load: Initial Load VEE Group - Commercial Interval (COM_INTD_MCS)

Measuring Component Type: COMSCALAR

- Description: Commercial Scalar Register
- Measuring Component Business Object: CM-ScalarValRegister
- Measurement Business Object: CM-FinalMeasurement
- Service Type: Electric
- Value Identifiers:

Value Identifier Type	Short-Hand Description	UOM
Measurement	KWH	KWH

- Valid VEE Groups:
  - Initial Load VEE Group - Scalar (SCALAR_MCS)
- Fallback VEE Groups:
  - Initial Load: Initial Load VEE Group - Scalar (SCALAR_MCS)

Measuring Component Type: INDINTERVALKVARH

- Description: Industrial KVARH Interval Channel
- Business Object: CM-IntervalChannel
- Measurement Business Object: CM-FinalMeasurement
- Service Type: Electric
- Value Identifiers:

Value Identifier Type	Short-Hand Description	UOM
Measurement	KVARH	KVARH

- Valid VEE Groups:
  - Initial Load VEE Group - Industrial Interval (IND_INTD_MCS)
- Fallback VEE Groups:
  - Initial Load: Initial Load VEE Group - Industrial Interval (IND_INTD_MCS)

Measuring Component Type: INDINTERVALKWH (Industrial KWH Interval Channel)

- Description: Industrial KWH Interval Channel
- Business Object: CM-IntervalChannel
- Measurement Business Object: CM-FinalMeasurement
- Service Type: Electric
- Value Identifiers:

Value Identifier Type	Short-Hand Description	UOM
Measurement	KWH	KWH

- Valid VEE Groups:
  - Initial Load VEE Group - Industrial Interval (IND_INTD_MCS)

- Fallback VEE Groups:
  - Initial Load: Initial Load VEE Group - Industrial Interval (IND_INTD_MCS)

TOU Map Templates and TOU Map Types

This section provides additional details for the TOU map templates and TOU map types listed above. In the case where different types of industrial customers use different rules for On Peak vs. Off Peak hours, different TOU map templates could be created for different TOU schedules. In this case, the primary difference between TOU map templates would be in the TOU Schedule Section.

TOU Map Template: IND_TOU_SCHED_1

- Description: TOU Schedule 1
- TOU Group: On-Peak / Off Peak TOU Group (ON-OFF_TOU_GRP)
- Default TOU: Off Peak (OFFPEAK)
- Work Calendar: Small Town USA Work Calendar (SMTUWRKCAL)
- Holiday TOU: Off Peak (OFFPEAK)
- Holiday Template: N/A
- Interval Size: 01:00:00
- TOU Schedule Section: Etc.

For each TOU map template, you might create multiple TOU map types for different interval sizes. For example:

TOU Map Type: IND_TOU_MAP_15_MIN

- Description: TOU Map Type - 15 Minutes (for Usage Calculation)
- TOU Map Business Object: D2-TOUMap
- Time Zone: US - Eastern Time
- Interval Size: 00:15:00
- Default TOU Map Template: TOU Schedule 1 (IND_TOU_SCHED_1)
- Override TOU Map Templates: N/A

TOU Map Type: IND_TOU_MAP_30_MIN

- Description: TOU Map Type - 30 Minutes (for Aggregation)
- TOU Map Business Object: D2-TOUMap
- Time Zone: US - Eastern Time
- Interval Size: 00:30:00
- Default TOU Map Template: TOU Schedule 1 (IND_TOU_SCHED_1)
- Override TOU Map Templates: N/A

TOU Map Type: IND_TOU_MAP_60_MIN

- Description: TOU Map Type - 60 Minutes (for Aggregation)
- TOU Map Business Object: D2-TOUMap
- Time Zone: US - Eastern Time
- Interval Size: 01:00:00
- Default TOU Map Template: TOU Schedule 1 (IND_TOU_SCHED_1)
- Override TOU Map Templates: N/A

Usage Groups and Rules

The table below lists the usage groups and corresponding usage rules for this implementation.

Usage Group	Usage Rules
RES_USAGE_GRP (Residential Usage Rules) Valid Device Configuration Types: • RESDVCCFG (Residential Device Configurations)	• RES_SCALAR_DETAILS (Retrieve KWH for Scalar Register)
COM_USAGE_GRP (Commercial Usage Rules) Valid Device Configuration Types: • COMDVCCFG (Commercial Device Configurations)	• COM_APPLY_MATH_KWH (Calculate KWH for Interval Channel) • COM_APPLY_MATH_KW (Calculate KW for Interval Channel)
IND_USAGE_GRP (Residential Usage Rules) Valid Device Configuration Types: • INDDVCCFG (Industrial Device Configurations)	• IND_GET_TOU_DATA (Calculate TOU-based KWH for interval channel) • CALC_KVA (Calculate KVA from KWH and KVARH interval channels)

Usage Subscription Types

This section provides additional details for each of the subscription types listed above.

Usage Subscription Type: RESIDENTIAL

- Description: Residential
- Usage Subscription Business Object: CM-ResidentialUS
- Usage Recipient: OUCCB (Oracle Utilities Customer Care and Billing)
- Valid Service Point Types:
  - Residential (RESIDENTIAL)
- Valid Usage Recipients:
  - OUCCB (Oracle Utilities Customer Care and Billing)
- Valid Usage Groups:
  - Residential Usage Rules (RES_USAGE_GRP)
- Fallback Usage Groups:

Effective Date	Usage Group
08-01-2010	Residential Usage Rules (RES_USAGE_GRP)

Usage Subscription Type: COMMERCIAL

- Description: Commercial
- Usage Subscription Business Object: CM-CommercialUS
- Usage Recipient: OUCCB (Oracle Utilities Customer Care and Billing)
- Valid Service Point Types:
  - Commercial (COMMERCIAL)
- Valid Usage Recipients:
  - OUCCB (Oracle Utilities Customer Care and Billing)
- Valid Usage Groups:

    - Commercial Usage Rules (COM_USAGE_GRP)
- Fallback Usage Groups:

Effective Date	Usage Group
08-01-2010	Commercial Usage Rules (COM_USAGE_GRP)

Usage Subscription Type: INDUSTRIAL

- Description: Industrial
- Usage Subscription Business Object: CM-IndustrialUS
- Usage Recipient: OUCCB (Oracle Utilities Customer Care and Billing)
- Valid Service Point Types:
    - Industrial (INDUSTRIAL)
- Valid Usage Recipients:
    - OUCCB (Oracle Utilities Customer Care and Billing)
- Valid Usage Groups:
    - Industrial Usage Rules (IND_USAGE_GRP)
- Fallback Usage Groups:

Effective Date	Usage Group
08-01-2010	Industrial Usage Rules (IND_USAGE_GRP)

Create Master Data

In this last step, we would create the actual master data (individual devices, service points, etc.) for the implementation. For purposes of this section, only a single example of each type of data is presented.

Contacts

A typical residential contact might look like this:

Residential - Person:

- Information: John Smith / 555-555-5555
- Contact Type: Residential - Person (RESIDENTIAL)
- Name: John Smith
- Home Phone: 555-555-5555

Service Points

A typical commercial service point might look like this:

Commercial Service Point:

- Information: 35 York Street, Burlington, MA, 01803, US / Commercial / Active
- Service Point Type: Commercial (COMMERCIAL)
- Status: Active
- Time Zone: US - Eastern Time
- Market: Small Town USA

- Main Contact: Phillip Jones
- Address:
  - Country: United States
  - Postal Code: 01803
  - Street Address: 35 York Street
  - City: Burlington
  - State: MA
- Measurement Cycle:
  - Measurement Cycle: Commercial Cycle 01 (COMMC01)
  - Route: Route 2
  - Sequence: 10

Devices

A typical industrial device might look like this:

Industrial Device:

- Information: 123456 / Industrial Device / Install Date/Time: 08-01-2010 12:00 AM / Pending / MeterTech, Inc. / Active
- Device Type: Industrial Device (INDDVC)
- Serial Number: 123456
- Internal Meter Number: 654321
- Pallet Number: 123456
- Manufacturer: MeterTech, Inc.
- Model: IND2010
- Incoming Data Shift: Shifted
- Arming Required: Arming Required
- Head-End System: MeterTech, Inc. (METERTECH)
- Status: Active

Device Configurations

A typical industrial device configuration might look like this:

Industrial Device Configuration:

- Information: Industrial Device / Effective Date/Time: 08-01-2010 12:00 AM / Industrial Device Configuration / 2 Measuring Component(s) / Active
- Device Configuration Type: Industrial Device Configuration (INDDVCCFG)
- Device: 123456 / Industrial Device / Install Date/Time: 08-01-2010 12:00 AM / Pending / MeterTech, Inc. / Active
- Effective Date/Time: 08-01-2010 12:00 AM
- Time Zone: US - Eastern Time
- Status: Active

Measuring Components

Typical industrial measuring components might look like this:

KVARH Interval Channel:

- Information: 123456 / 2 / Industrial KVARH Interval Channel
- Measuring Component Type: Industrial KVARH Interval Channel (INDINTERVALKVARH)
- Device Configuration: Industrial Device / Effective Date/Time: 08-01-2010 12:00 AM / Industrial Device Configuration / 2 Measuring Component(s) / Active
- Consumption Reference Measuring Component: N/A
- How to Use: Consumptive
- Number of Digits Left: 5
- Number of Digits Right: 2
- Channel Multiplier: 1.000
- Latest Read Date/Time: N/A
- Channel ID: 2

KWH Interval Channel:

- Information: 123456 / 1 / Industrial KWH Interval Channel
- Measuring Component Type: Industrial KWH Interval Channel (INDINTERVALKWH)
- Device Configuration: Industrial Device / Effective Date/Time: 08-01-2010 12:00 AM / Industrial Device Configuration / 2 Measuring Component(s) / Active
- Consumption Reference Measuring Component: N/A
- How to Use: Consumptive
- Number of Digits Left: 5
- Number of Digits Right: 2
- Channel Multiplier: 1.000
- Latest Read Date/Time: N/A
- Channel ID: 1

Install Events

A typical industrial installation event might look like this:

Industrial Install Event:

- Information: Install Date/Time: 08-01-2010 / On
- Device Configuration: Industrial Device / Effective Date/Time: 08-01-2010 12:00 AM / Industrial Device Configuration / 2 Measuring Component(s) / Active
- Service Point: 47 North Street, Burlington, MA, 01803, US / Industrial / Active
- Status: On
- Installation Date/Time: 08-01-2010
- Installation Constant: 1.00000
- Device On/Off Status: On
- On/Off History: N/A

TOU Maps

A typical industrial TOU map might look like this:

- Information: Industrial TOU Map
- TOU Map Type: TOU Map Type - 15 Minutes (IND_TOU_MAP_15_MIN)
- Status: Active
- Status Reason: N/A
- Override TOU Map Template: N/A
- Dynamic Option / Dynamic TOU Map Section: N/A
- TOU Data List:
    - 08-01-2010 00:15 AM - Off Peak
    - 08-01-2010 00:30 AM - Off Peak
    - 08-01-2010 00:45 AM - Off Peak
    - 08-01-2010 01:00 AM - Off Peak
    - Etc.

Usage Subscriptions

A typical commercial usage subscription might look like this:

Commercial Usage Subscription:

- Information: Commercial / 08-01-2010 12:00 AM / Active
- Usage Subscription Type: Commercial (COMMERCIAL)
- Status: Active
- Start Date/Time: 08-01-2010 12:00 AM
- End Date/Time: 08-01-2020 12:00 AM
- Usage Recipient: Oracle Utilities Customer Care and Billing (OUCCB)
- Usage Approval: Not Required
- External ID:
- Main Contact: Phillip Jones / Business Phone: 555-555-5555
- Time Zone: US - Eastern Time
- Factor Overrides: N/A
- Usage Groups:

Effective Date/Time	Expiration Date/Time	Usage Group
08-01-2010 12:00 AM	08-01-2011 12:00 AM	Commercial Usage Rules (COM_USAGE_GRP)

- Service Points:

Service Point	Start Date/Time	Stop Date/Time	Usage	Use Percent
35 York Street, Burlington, MA, 01803, US / Commercial / Active	08-01-2010 12:00 AM	08-01-2011 12:00 AM	Add	100

Appendix A

Measurement Services

This appendix provides brief descriptions of the base package measurement services used by VEE rules and measurement functions. The measurement services described in this appendix are implemented as business services. These business services can be used by custom algorithms or BPA/Service scripts created for your implementation.

The table below lists the available back package measurement services, including the business service that implements each service and a brief description for each.

Measurement Service	Business Service	Description
Add Scalar Value To Intervals	D1-AddScalarValueToIntervals	Uses the Apply Formula measurement service to add a scalar value to the value of a specified set of interval data.
Adjust Intervals to Supplied Value	D1-AdjustIntervalsToSuppldVal	Uses the Apply Formula measurement service to adjust the total value of a specified set of interval data to a scalar value.
Apply Formula	D1-ApplyFormula	Used to apply a formula to a specified set of interval data, either by applying a summary function against all intervals of the set, or by manipulating each individual interval in series via a formula using declared constants, or within the context of other sets of input interval data.
Apply TOU Map To Interval Measuring Component	D1-ApplyTOUMapToIntervalMC	Used to apply a TOU map to a set of intervals for a specified measuring component and date/time range, thereby isolating and summarizing those intervals that occurred during a specific time of use.
Axis Conversion	D1-AxisConversion	Used to convert interval data between units of measure (UOMs) and interval sizes (SPIs), including the conversion between peak and consumption-oriented UOMs.
Convert Scalar Consumption To Interval	D1-IntervalizeScalarConsumptn	Used to convert a scalar consumption value to a set of interval measurements.

Measurement Service	Business Service	Description
Create Intervals	D1-CreateIntervals	Used to create interval data based on supplied parameters (UOM, SPI, number of intervals, value, etc.)
Divide Intervals By Scalar Value	D1-DivideIntervalsByScalarVal	Uses the Apply Formula measurement service to divide the values of a specified set of interval data by a scalar value.
Extract Subset of Intervals	D1-ExtractSubsetOfIntervals	Used to extract a subset of interval data from a specified set of intervals.
Identify Spikes	D1-IdentifySpikes	Used to identify spikes in a specified set of interval data based on a spike percentage tolerance.
Insert Intervals	D1-InsertIntervals	Used to insert one or more intervals into a set of interval measurements.
Merge Intervals	D1-MergeIntervals	Used to merge a subset of interval data with a specified set of intervals (where overlaps occur, the subset intervals replace the original intervals).
Multiply Intervals By Scalar Value	D1-MultiplyIntervalsByScalarVal	Uses the Apply Formula measurement service to multiply the values of a specified set of interval data by a scalar value.
Remove Intervals	D1-RemoveIntervals	Used to remove one or more intervals from a set of interval measurements.
Retrieve Interval Consumption	D1-IntvConsumptionRetriever	Used to retrieve one or more interval measurements.
Retrieve Scalar Consumption	D1-ScalarConsumptionRetriever	Used to retrieve one or more scalar measurements.
Set Condition	D1-SetCondition	Used to set the condition (status) code of a specified set of interval data.
Shift Intervals	D1-ShiftIntervals	Used to shift one or more intervals forward or backward in time.
Subtract Scalar Value From Intervals	D1-SubtractScalarValToIntervls	Uses the Apply Formula measurement service to subtract a scalar value from the value of a specified set of interval data.

Use the Business Service portal to view more details concerning these measurement services.

Appendix B

Glossary

This glossary provides definitions of commonly used terms.

360 Degree View - Audit View

A zone that allows users to view an interval measurement curve for a given period overlaid with the count of audit records for each individual measurement. It also allows users to magnify a portion of the curve and see how the measurements looked at different points in time.

360 Degree View - Final Values Overlay

A zone that graphs final measurements for a measuring component, and provides the ability to overlay the graphed data with final measurements from other measuring components. The zone also permits overlaying data from the same measuring component for different time periods, as well as data from measuring components measuring different quantities, such as temperature.

360 Degree View - Time of Use by Day

A zone that displays daily TOU-mapped usage data for a measuring component based on a user-defined time period and TOU Map.

360 Degree View - Time of Use Overlay

A zone that displays an overlay of the TOU periods on a final measurement along with totalized TOU consumption based on a user-defined time period and TOU map.

360 Degree View - Timeline Zone

A zone that displays a timeline of activities and other events for a given service point, device, etc.

Activity Type

Defines properties common to a specific type of activity.

Advanced Metering Infrastructure (AMI)

Refers to systems that measure, collect and analyze energy usage, and interact with advanced devices such as electricity meters, gas meters, heat meters, and water meters, through various communication media either on request (on-demand) or on pre-defined schedules.

Measurement Service - Add Scalar Value To Intervals

Measurement service that uses the Apply Formula measurement service to add a scalar value to the value of a specified set of interval data.

Adjust Intervals To Scalar Quantity

An initial measurement data function used to adjust all interval values within a user-defined time period in an interval measurement such that the total of all interval values equals a user-defined scalar quantity. This function results in a measurement that retains the shape of the original measurement, but has been scaled up or down.

Adjust Intervals To Supplied Value

Measurement service that uses the Apply Formula measurement service to adjust the total value of a specified set of interval data to a scalar value.

Adjust Intervals Using Math

A measuring component consumption function used to adjust all interval values within a user-defined time period using math operations (add, subtract, multiply, or divide).

Aggregation

Measurements that represent an summarization of other measurements from a potentially diverse set of devices. For example, an aggregation may derive the sum of the electric energy of all residential customers in a particular postal code within the utility's service territory.

Aggregator

A class of measuring component that stores measurements that represent an summarization of other measurements from a potentially diverse set of devices. For example, an aggregator may derive the sum of the natural gas consumption of all residential customers in a particular postal code within the utility's service territory.

Apply Formula

Measurement service used to apply a formula to a specified set of interval data, either by applying a summary function against all intervals of the set, or by manipulating each individual interval in series via a formula using declared constants, or within the context of other sets of input interval data.

Apply TOU Map To Interval Measuring Component

Measurement service used to apply a TOU map to a set of intervals for a specified date/time range, thereby isolating and summarizing those intervals that occurred during a specific time of use.

Automatic Meter Reading (AMR)

The technology of automatically collecting consumption, diagnostic, and status data from water meter or energy metering devices (water, gas, electric) and transferring that data to a central database for billing, troubleshooting, and analyzing.

Axis Conversion

Measurement service used to convert interval data between units of measure (UOMs) and interval sizes (SPIs), including the conversion between peak and consumption-oriented UOMs.

Bill Determinants

Measurement data summarized for use by a billing application. Bill determinants can take the form of TOU-mapped interval consumption, scalar consumption, scalar readings, and/or interval consumption obtained via measurements. A common variety of bill determinant is TOU-mapped interval consumption, which reduces a full month's worth of interval data into several buckets of consumption based on time of use.

Command

A communication sent to a device to perform some action on the device, such as Connect, Disconnect, Commission, Decommission, On-Demand Read, or Device Status Check (Ping)

Communication

A record of a message sent between Oracle Utilities Smart Grid Gateway and an external system, such as a head-end system or edge application. Communications can flow both inbound and outbound, and can be both one-way and two-way.

Completion Event

Records used to create or update transactions that reflect the effect of an activity. For example, issuing a commission device command could result in the creation or update of an install event while a read device command could result in the creation of initial measurement data.

Consumption

A measurement by a given device of the amount of energy, water, gas, etc. consumed over a given time period. Synonymous with the term "measurement".

Consumptive

Describes a measuring component for which readings are equivalent to the consumption. For example, if we receive a reading of 400 on January 15 and a reading of 600 on February 15, a consumptive measuring component's consumption between January 15 and February 15 would be 600 (not 200).

Contact

An individual or a business entity with which a company has contact. Each contact must reference a contact type.

Contact - Email

Email addresses related to a contact

Contact - Identifier

Identifiers related to a contact, such as social security number, driver's license number, or the contact's ID in a prior system.

Contact - Name

Names related to a contact

Contact - Phone

Phone numbers related to a contact

Contact Type

Defines the properties of a class of entities (businesses, persons).

Convert Scalar to Interval Consumption

A measuring component consumption function used to convert scalar consumption into interval consumption. The converted consumption is held in a new scratchpad measuring component. The "shape" of the new interval measurement can be based on either a profile or can have a flat distribution.

Convert Scalar Consumption To Interval

Measurement service used to convert a scalar consumption value to a set of interval measurements.

Create Intervals

Measurement service used to create interval data based on supplied parameters (UOM, SPI, number of intervals, value, etc.)

Create Scratchpad Interval Consumption

A measuring component consumption function used save all or some of the final measurements shown in the zone as a new initial measurement for a scratchpad measuring component.

Create/Override

A measuring component consumption function used to create new initial measurement data for a selected measuring component for all or part of a selected time period. This function can either copy existing final measurements (for example, from a profile) or create new measurement data for the IMD it creates.

Demand

The rate at which a commodity is delivered at a given instant or averaged over a designated time. For electricity, demand is often expressed in kilowatts (kW) or kilovolt-amperes (kVa).

Device

A physical or virtual object that holds one or more measuring components that can produce data to be handled by the system. Devices can include meters, substations, transformers, demand response devices, weather stations, etc.

Device Configuration

A specific configuration of a device. Over time, a device can have many configurations. Use of effective-dated device configuration allows the device to retain its identifier(s) even while the quantities it is measuring are changing.

Device Configuration Type

Defines the properties of device configurations of this type, including the valid types of measuring components that can be configured for the device.

Device Event

An event of some sort that has taken place relative to a device. Device events can include power outages, power restorations, tampering alerts, command completion, and other information

Device Status Check

A communication sent to a device to test whether the device is communicating with the network, determine the connection status of the meter, and when possible if there are any known malfunctions

Device Type

Information about a class of devices, including properties that apply to all devices of a type, but can be overridden for an individual device.

Distribution Company (DISCO)

A utility company that constructs and maintains the distribution network that delivers a commodity to customers. Depending upon the regulations within the territory, a distribution company may or may not be responsible for billing the customer.

Divide Intervals By Scalar Value

Measurement service that uses the Apply Formula measurement service to divide the values of a specified set of interval data by a scalar value.

Dynamic Option

Used to specify terms that override how usage is normally calculated - such as a critical peak period that affects the TOU mapping of interval consumption.

Dynamic Option Event

The period of time during which a dynamic option is applicable. A dynamic option may have many events over time.

Dynamic Option Type

Used to define information common to dynamic options of a specific type.

Exception Type

Defines properties common to many exceptions, including the category of the exception.

Extract Subset of Intervals

SMeasurement service used to extract a subset of interval data from a specified set of intervals.

Factor

A centrally stored set of values for use in validation rules, bill determinants calculations, and other processes. A factor can have different values depending upon some definable attribute of a system object, such as customer size associated with a service point. The values are effective-dated so that changes over time are retained. Examples of factors can include minimum/ maximum thresholds, loss factors, etc. Classes of factors are defined that can have numeric values (as in the above examples), or values pointing to profile measuring components or VEE groups.

Factor Value

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

Final Measurement

Measurement data that has been validated, and if necessary, edited & estimated, and is ready for use in down-stream processing such as bill determinants calculations. Only one final measurement can exist for a given date/time for a given measuring component; one final measurement exists per interval, and likewise one final measurement exists for each scalar reading. In both cases, the final measurement value stored represents the amount consumed between its date/time and the prior final measurement's date/time.

Function

An online-initiated action applied to measurement data, comprising one or more measurement services.

Head-End System

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

Identifiers

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

Identify Spikes

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

Inbound Communication

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

Initial Measurement Data Function (IMDF)

An online means of manipulating initial measurement data. IMD Functions typically move intervals (shift, insert, remove, etc.) or manipulate their values or conditions.

Insert Intervals

An initial measurement data function used to insert intervals into initial measurement data. Intervals can be inserted at either the beginning or end of the measurement. The end date/time of the measurement is shifted to account for the inserted intervals.

Insert Intervals (service)

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

Inbound Communication

Communication sent to MDF (Meter Data Framework) 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 (Meter Data Framework). Over its lifecycle (as pertains to MDM - Meter Data Management), any readings within the IMD are converted into consumption, which is then typically subject to VEE processing and then finalized - meaning stored as final measurements. Only initial measurements can be edited directly by end users of MDM. An IMD for a scalar measuring component will have a single measurement (along with a reading from which the measurement value is derived), while an IMD for an interval measuring component will usually contain multiple interval measurements.

Installation Constant

An installation constant is set to a value other than 1 as an indication that when calculating consumption, the installation requires that measurement data be multiplied by this value to get accurate results.

Installation Event

A device's installation information at a service point. The install event represents both the installation and removal of a device. It also records turning a device on or off while it is installed at a service point.

Installation On and Off History

A single installation event records each time the device is turned on and turned off while it is installed at a service point.

Interval Channel (Measuring Component)

A business object (BO) that represents channels associated to a device.

Interval Channel Type - Physical (Measuring Component Type)

A business object (BO) that maps properties of interval measuring component types for those Measuring Components that are part of physical devices.

Interval Channel Type - Scratchpad (Measuring Component Type)

A business object (BO) that maps properties relevant to stand-alone measuring components functioning as scratchpads for interval data manipulation.

Interval Data

Time-series data in which measurements are captured in pre-defined intervals (5 minutes, 15 minutes, 1 hour, etc.). A set of interval measurements for an interval measuring component composes an individual initial measurement data record.

Interval Data Services

Services used to access and manipulate interval measurements.

Interval Scratchpad (Measuring Component)

A stand-alone measuring component that provides the user with a means to manipulate measurement data without affecting existing measurements.

Interval Size

The "size" of an interval, representing the length of time between intervals. Interval size is typically measured in seconds-per-interval (SPI).

Manual Meter

A business object (BO) used to model a meter that does not accommodate two-way communications and must be read manually.

Manual Meter Installation Event

A business object (BO) that defines the lifecycle of the installation of a manual meter at a service point.

Manual Meter Type

A business object (BO) used to model properties for meters that are manually read.

Manufacturer

The company that makes devices, defined as an attribute of the device itself.

Market

The jurisdiction or regulatory environment in which a service point participates, defining the valid service providers and their roles. While each service point specifies only one market, different service points throughout the utility's service territory can be linked to different markets.

Market - Fallback Service Provider

For a given market relationship type, a fallback service provider may be defined at the market level, rather than storing the information redundantly on each service point. For example, an entire market might have only one ISO, and if the utility wants to store this information, they can identify the ISO as a fallback service provider for the market and the market relationship type of ISO.

Market - Relationship Type

The valid roles within a market (ISO, Distribution Company, Retailer, etc.) that have some business significance in the application.

Market - Valid Service Provider

The valid service providers for each market relationship type relevant for a given market. The service providers referenced on a service point must be valid for the combination of the service point's market and the market relationship type.

Market Participant

A variety of service provider; a company with a role within a given market such as a retailer or a distribution company.

Measuring Component Consumption Function (MCCF)

An online means of initiating the process of adding or editing measurements. Measuring Component Consumption Functions typically create new initial measurements based on a copy of existing final measurements.

Measurement

A measurement in MDM is synonymous with consumption, which implies that constants or multipliers may have been applied to its value. This term can be used in the context of an IMD or in reference to Final Measurements.

Measurement Condition

Codes that indicate the circumstances (estimated, missing, etc.) of individual measurements. Conditions are assigned to both scalar and interval measurement data both for initial measurement data and final measurements.

Measurement Cycle

The measurement cycle can serve two purposes: it can define the schedule for manual meter reading of devices at service points in that cycle, and it can also be configured to define when to create usage transactions for usage subscriptions associated to service points in the cycle.

Measurement Cycle Route

The route used to collect measurements for a given measurement cycle.

Measurement Cycle Route Sequence

The sequence in which measurements are collected along a measurement route.

Measurement Cycle Schedule

Defines the dates on which devices are scheduled to be read.

Measurement Service

Java services that can be invoked to manipulate interval and scalar measurements. Measurement services are invoked by measurement functions (available through certain zones within MDM), and are also used within processing of usage and VEE rules.

Measuring Component Summary

A zone shown on the VEE Group portal that displays a list of measuring components that reference a given VEE group.

Measuring Component

A single point for which data will be received and stored in the system. A measuring component can be associated to a physical device, which can have one or more measuring components, or it can be stand-alone, meaning that it is not associated to a physical device (for example, an aggregator or interval scratchpad).

Measuring Component Type

The definition of the most important properties of a measuring component, including what it measures, how regularly it measures it, whether it should be connected to a physical device or if it's used as a scratchpad or an aggregator, how its final measurements should be stored and how its user-defined values should be calculated, what rules govern VEE for Measuring Components of the type, as well as numerous display properties that are relevant within MDM. The measuring component type also defines sets of valid attribute values for groups of measuring components belonging to the type.

Measuring Component Types Referencing Group

A zone shown on the VEE Group portal that displays a list of Measuring Component types that reference the VEE group being viewed.

Merge Intervals

Measurement service used to merge a subset of interval data with a specified set of intervals (where overlaps occur, the subset intervals replace the original intervals).

Meter

A device used to measure a quantity of a service (electricity, gas, etc.) delivered to a service point.

Meter Read Download Activity Type

The structure and business rules applicable to downloading meter read information onto a handheld device.

Model

A specific model of a device produced by a manufacturer. Models for a single manufacturer can have diverse service types.

Multiplier

A value that may be applied to adjust the consumption values calculated for a device. Examples include meter/device multiplier, installation constant, loss factor, etc.

Multiply Intervals By Scalar Value

Measurement service that uses the Apply Formula measurement service to multiply the values of a specified set of interval data by a scalar value.

New Scalar Reading

A measuring component consumption function used to create new initial measurement data containing a reading (rather than consumption) for the scalar measuring component displayed in the zone for a user-defined time period. A reading refers to the measurements as read from the meter, while consumption refers to the total consumption, accounting for meter multiplier and/or offset.

Normalized storage

Storing measurement data in a manner that allows for aggregation and reporting of data through database logic (SQL). Applies to both scalar and interval measurements.

Off-Peak Period

A time period during which the least amount of some consumable is being used. OR A period of relatively low system demand as specified by the supplier.

On-Peak Period

A time period during which the greatest quantity of some consumable is being used OR A period of relatively high system demand as specified by the supplier.

One-Way Communication

Communication from head-end system to Oracle Utilities Smart Grid Gateway that does not trigger a response. Examples of one-way communications include usage readings and device events.

Oracle Utilities Meter Data Management

Oracle Utilities application that provides functionality for handling large volumes of meter data to enable increased accuracy, flexibility, and scalability.

Oracle Utilities Smart Grid Gateway

Oracle Utilities application that provides functionality for orchestrating communication with head-end systems to support import of usage and events, and issuing of meter commands.

Outbound Communication

Communication sent from Oracle Utilities Smart Grid Gateway to a head-end system or other external system.

Peak

The maximum value for some measurable quantity recorded over a specified time period. A measuring component that measures peak quantities will record the highest value for the quantity over a period of time.

Peak Demand

The maximum rate of commodity consumption over a specific period of time.

Processing Method

Methods used to define the format or means by which a service provider receives data from the application, such as bill determinants, interval data, or meter events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and usage transactions. Processing methods can also be used to define the information an external system wishes to subscribe to receive from our application. A BO or batch extract code are the typical processing methods defined for the transmission of data to a service provider.

Processing Role

Each processing method has a processing role, which defines the purpose of the processing method. Some examples of processing roles include: * Initial Measurement Creation (D1) * Device Activity Notification (D1) * Usage Transaction Notification (D2) * Usage Transaction Creation (D2)

Profiling of Scalar Data

The process of applying an interval consumption "shape" to a scalar measurement, using an existing interval measuring component. The individual interval values are adjusted such that when totaled, they equal the value of the scalar measurement.

Reading

The value recorded by a measuring component at a given point in time. A reading often needs to be interpreted in the context of an earlier reading in order to derive a consumption value that would be stored as a measurement. For example, a reading of 1000 for a subtractive measuring component taken on February 1 in the context of a prior reading of 600 taken on January 15 would result in a consumption (measurement) of 400. Readings can either be consumptive or subtractive.

Register (Measuring Component)

A business object (BO) that represents a scalar register found on a standard or smart meter. It does not have a lifecycle, and should be associated with a device configuration.

Register Type - Physical (Measuring Component Type)

Measuring component type business object (BO) that enumerates the properties used by scalar registers.

Remove Intervals

An initial measurement data function used to remove intervals from initial measurement data. Intervals can be removed from either the beginning or end of the measurement. The end date/time of the measurement is shifted to account for the new number of intervals in the measurement.

Remove Intervals (service)

Measurement service used to remove one or more intervals from a set of interval measurements.

Retail Company

A company that is authorized to buy and re-sell a commodity (such as electricity or gas) directly to customers based on territorial regulations.

Retrieve Interval Consumption

Measurement service used to retrieve one or more interval measurements.

Retrieve Scalar Consumption

Measurement service used to retrieve one or more scalar measurements.

Save As Interval Consumption

A measuring component consumption function used save all or some of the final measurements shown in the zone as a new initial measurement for a different (new or existing) measuring component.

Scalar Usage

A measurement of the amount of energy, water, gas, etc. consumed for a given measuring component for a given time period.

Seconds Per Interval

Seconds Per Interval, a way of expressing the length of time between which measurements are taken.

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 Type

Specific types of service, such as electric, gas, steam, etc.

Set Condition Codes

An initial measurement data function used to set condition codes within initial measurement data. Condition codes indicate the circumstances (estimated, missing, etc.) of individual measurements. Condition codes are assigned to both scalar and interval measurement data both for initial measurement data and final measurements.

Set Condition

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

Shift Intervals

A function used to shift intervals of initial measurement data forward or backward time. The end result is a measurement with a different start or end time.

Shift Intervals (service)

Measurement service used to shift one or more intervals forward or backward in time.

Smart Meter

A business object (BO) used to model smart meters of different service types.

Smart Meter Installation Event

A business object (BO) that defines the lifecycle and rules for installing a smart meter at a service point.

Smart Meter Type

A business object (BO) for device type that references a head-end system as well as a collection of head-ends that are valid for devices of the type, and indicates whether incoming data incorporates the daylight savings time shift. Additionally, the smart meter type includes a list of valid device configurations for its devices.

Smooth Spikes

An initial measurement data function used to reduce "spike" intervals (intervals with values that are more than a user-defined percentage higher than other intervals within the initial measurement data). The function smooths spikes using linear interpolation as follows:

- Get the value of the interval immediately preceding the spike (the "Left Value").
- Get the value of the interval immediately after the spike (the "Right Value").
- Subtract the Left Value from the Right Value, divide result by two. The result is called the Estimation Adder
- Add the Estimation Adder to the interval immediately before the spike.

Substation

A subsidiary station of an electricity generation, transmission and distribution system where voltage is transformed from high to low or the reverse using transformers.

Subtract Scalar Value From Intervals

Measurement service that uses the Apply Formula measurement service to subtract a scalar value from the value of a specified set of interval data.

Subtractive

Describes a measuring component for which consecutive readings must be subtracted to derive a consumption value.

Time of Use

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

TOU Group

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

TOU Map

A collection of TOU map data derived via a given TOU map template at a specific interval size (TOU). A TOU map is typically specified when configuring a usage calculation rule for TOU mapping. This TOU map's data will then be used when summarizing the interval data for each TOU period.

TOU Map Data

An interval date/time and its associated TOU as defined by a TOU map template. For example, if the schedule defined for a TOU map template specifies that the period on weekdays from 9 AM to 5 PM falls into On-Peak, and the data is hourly, rows would be stored in the TOU map data table with the date/time 5/3/2010 at 10 AM, 5/3/2010 at 11 AM, 5/3/2010 at 12 PM, etc., each with a value of On-Peak.

TOU Map Template

The schedule used for TOU map data generation, for example defining year, month, and day ranges and which TOUs should be used during each.

TOU Map Type

Defines certain important properties of TOU maps of the type, including the interval size (SPI) and the valid TOU map templates.

Transformer

A device that transfers electrical energy from one circuit to another.

Two-Way Communication

Communication sent from Oracle Utilities Smart Grid Gateway to an external system, such as a head-end system or edge application that triggers a response. Most commands are two-way communications, where Oracle Utilities Smart Grid Gateway issues a command, and the head-end system sends a response as to the success or failure of the command.

Unit of Measure

Identifies quantities measured, such as KWH, KW, cubic feet, degrees Celsius, etc.

Usage

A generic term for the amount of energy, water, gas, etc. consumed at one or more service points, sometimes representing quantities that have been adjusted from the original calculated consumption.

Usage Calculation Group

A set of sequenced usage rules used to calculate usage for a usage subscription.

Usage Rule

Business rules / logic used to calculate usage (bill determinants), such as a TOU-mapped consumption calculation. Each rule is a modular unit that can be grouped together and sequenced within a calculation group.

Usage Rule - Apply Math (Interval Data)

A usage rule used to summarize interval measurements and apply a mathematical formula against the results to derive a usage quantity.

Usage Rule - Get Interval Data

A usage rule used to get interval data for a measuring component and date range.

Usage Rule - Get Scalar Details

A usage rule used to assemble scalar readings and measurements (consumption).

Usage Rule - Get TOU Mapped Usage

A usage rule used to summarize interval measurements into TOU buckets based on a TOU map (where the TOU map is created based on the schedule defined within a TOU map template).

Usage Rule - Math

A usage rule used to derive a curve given a formula, apply TOU mapping to a derived curve, and perform mathematical operations on Usage Transaction Service Quantity entries.

Usage Rule - Validate Usage Against Tolerance

A usage rule used to validate calculated usage against a specified tolerance.

Usage Rule Eligibility Criteria

Configured criteria used to determine whether to execute a specific usage rule when calculating usage.

Usage Subscription

A record of an ongoing request to send one or more service points' usage to one or more external systems (such as a billing application).

Usage Subscription Type

A collection of properties defining a class of usage subscriptions. The usage subscription type also controls valid values for various attributes of usage subscriptions.

Usage Transaction

A record of bill determinant calculations for a usage subscription.

User-Defined Measurement Values

Additional values optionally stored with a given measurement that can be used in various calculations. For example, a customer's gas consumption might be measured in cubic feet, but needs to be sent to a billing system in therms. A user-defined value to convert consumption in cubic feet into therms can be configured, and the therm value will then be stored with the measurement in cubic feet.

Validation, Estimation, and Editing (VEE)

The process by which initial measurement data is validated, estimated (if necessary) and edited (if necessary) based on a set of user-defined rules.

VEE Eligibility Criteria

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

VEE Exception

An exception generated during Validation, Estimation and Editing (VEE) processing of initial measurement data. Exceptions are assigned a severity that is used in determining whether or not the initial measurement data should be transitioned into an exception state.

VEE Group

A collection of VEE Rules.

VEE Group Matrix (Factor)

A VEE rule within a VEE group can be configured to pick from a list of VEE groups (referred to as a matrix) whose rules to execute next. This list of VEE groups is configured as the values of a factor. One example of its use could be to call geographically-specific VEE groups from within a larger-purpose group. A residential VEE group might contain a rule that will pick the VEE group to execute based on service point location, where the VEE Group Matrix specifies: SP in the North - VEE Group N SP in the East - VEE Group E SP in the South - VEE Group S

VEE Group Matrix (Factor) Referencing Group

A zone that displays a list of VEE group matrices (factors) that reference the VEE group being viewed in the VEE group portal.

VEE Rule

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

VEE Rule - Ensure IMD Exists for Sibling MCs

VEE Rule that checks to ensure that initial measurement data exists for siblings measuring components.

VEE Rule - High/Low Check

VEE Rule that compares consumption of incoming initial measurement data against historical data as a means of assuring the reasonableness of the data.

VEE Rule - Interval Adjustment from Scalar

VEE rule that adjusts initial measurement data for interval readings based on an existing scalar value

VEE Rule - Interval Averaging Estimation

VEE rule that estimates missing interval values based on averaging of historical data for the same device and measuring component.

VEE Rule - Interval Interpolation Estimation

VEE rule that estimates missing interval values based on linear interpolation.

VEE Rule - Interval Profile Estimation

VEE rule that estimates missing interval values based on an associated profile measuring component.

VEE Rule - Interval Replacement Rule

VEE Rule that determines whether or not an incoming initial measurement will replace any existing measurements and if so, whether or not to allow this to happen.

VEE Rule - Interval Size Validation

VEE Rule that checks to ensure that the interval size of the incoming initial measurement data matches the value defined in the measuring component type.

VEE Rule - Interval Spike Check

VEE Rule that examines incoming initial measurement data to identify intervals with conspicuously high usage relative to surrounding intervals.

VEE Rule - Multiplier Check

VEE Rule that checks to ensure that the device multiplier value of the incoming initial measurement data matches the multiplier value stored on the measuring component.

VEE Rule - Negative Consumption

VEE Rule that checks for negative consumption.

VEE Rule - Raise Missing Quantity Exception

VEE rule that creates an exception if a specified percentage of measurements lies within the range of condition flag values for missing but not within the range of values for outage.

VEE Rule - Scalar Calculation from Interval

VEE rule that calculates a scalar value from the interval data values from the same date range

VEE Rule - Scalar Estimation

VEE rule that estimates missing scalar readings based on averaging of historical data for the same device and measuring component.

VEE Rule - Scalar Profile Estimation

VEE rule that estimates missing scalar readings based on an associated profile measuring component.

VEE Rule - Scalar Replacement Rule

VEE Rule that identifies whether or not a scalar reading will completely replace an existing measurement and if so, whether or not to allow this to happen.

VEE Rule - Sum Check

VEE Rule that checks the difference between the total consumption of incoming initial measurement data against the total consumption (based on final measurements) for one or more related measuring components on the same device over the same time period.

VEE Rule - Unit Of Measure Check

VEE Rule that checks to ensure that the Unit of Measure (UOM) of the incoming initial measurement data matches the UOM specified on the measuring component's type.

VEE Rule - Zero Consumption Check

VEE rule that creates an exception if it detects zero consumption for the current initial measurement.

Appendix C

Base Package Configuration Objects

This appendix provides lists of some base package configuration objects that can aid when implementing Oracle Utilities Meter Data Management. This includes:

- **Meter Data Framework Base Package Data Areas**
- **Oracle Utilities Meter Data Management Back Package Data Areas**
- **Meter Data Framework Base Package Extendable Lookups**
- **Oracle Utilities Meter Data Management Back Package Extendable Lookups**

Meter Data Framework Base Package Data Areas

The table below lists the available meter data framework base package data areas.

Data Area	Description
D1-ActivityCommon	Activity Command Data Area
D1-ActivityTypeBasis	Common Schema of Activity Type
D1-AdminBOExceptionHandling	Admin BO Exception Handling Common
D1-AmountandCondition	Amount and Condition Data Area
D1-BOAuditChangedValues	BO Audit Changed Values
D1-CancelCommand Common	Cancel Command Data Area
D1-CommTypeBasicDA	Communication Type Basis
D1-CommandReqIBCommCommonDA	Common Inbound Commn for Command Request
D1-CommandReqOBCommCommonDA	Common Outbound Commn for Command Request
D1-CommandRequestCommon	Command Request Common Data Area
D1-CommandRequestTypeCommon	Command Request Type Common Data Area
D1-CommonActivityType	Activity Type
D1-CommonFactor	Common Factor
D1-CommonFactorValue	Common Factor Value
D1-CommonIMD	Common Initial Measurement Data
D1-CommonIMDAudit	Common IMD Audit
D1-CommonInstantiableBOOutput	Common Instantiable BO's Output
D1-CommonMeasurementSvcsOutput	Common Measurement Service Output
D1-CommonMeterData	Common Meter Data
D1-CommonMeterTypeData	Common Meter Type Data
D1-CommnoProcMethodDetails	Common Processing Method Details
D1-CommonSrvcProvDetails	Common Service Provider Details
D1-CommonSyncRequestContact	Common for Sync Request Contact
D1-CommonSyncRequestDC	Common for Sync Request Device Config
D1-CommonSyncRequestDvc	Common for Sync Request Device
D1-CommonSyncRequestIE	Common for Sync Request Install Event
D1-CommonSyncRequestInData	Sync Request Inbound Common Data
D1-CommonSyncRequestMC	Common for Sync Request Inbound MC
D1-CommonSyncRequestSP	Common for Sync Request Service Point

Data Area	Description
D1-CommonUomSpiIdentGroup	Common UOM/SPI Identification Group
D1-CommonVEEIMD	Common VEE Initial Measurement Data
D1-CompletionEvent	Completion Event Parent Data Data
D1-ConDisconNotification	ConnectDisconnect Notification
D1-DeviceConfigRelatedActivity	Device Config Related To Activity DA
D1-DeviceEvent	Device Event
D1-DeviceRelatedActivity	Device Related to Activity Data Area
D1-DeviceStatusCheckNotif	Device Status Check Notification
D1-EligibleProfileFactors	Eligible Profile Factors for Data Creation
D1-EventBarProfileData	Event Bar Profile Data
D1-EventBarProfileList	Event Bar Profile List
D1-EventInformation	Event Information
D1-ExtendableLookupColumn	Extendable Lookup Common Data Area
D1-ExtendedMsrmtList	Extended ML
D1-FVOverlayDefaultProfileList	Final Values Overlay Default Profile List
D1-FVOverlayProfileData	Final Values Overlay Profile Data
D1-FailReponse	Fail Response Data Area
D1-FallbackSPrValList	Fallback Service Provider Validation List
D1-FileHeaderInfoDA	Upload Statistics File Header Information
D1-GenSavePointDispDA	Generic Savepoint Dispatcher DA
D1-IBCommCommonDA	Inbound Communication Common
D1-IERelatedActivityDataArea	Common Schema of IE Related Activities
D1-InstallEventRelatedActivity	Activity Related Installation Event
D1-MCSnapShot	SnapShot group for Sync Request Inbound MC
D1-MRDownloadActivityCommon	Common schema Meter Read Dwnld Activity
D1-MeasurementDataList	Measurement Data List
D1-NumberOfIntervalsAndDateTm	Number of Intervals and Date Time
D1-OBCommCommonDA	Outbound Communication Common Data Area
D1-PayloadProcessErrorsDA	Payload Errors DA
D1-PayloadProcessInfoDA	Payload Processing Information
D1-ReadDeviceNotification	Read Device Notification Data Area
D1-ReceivedResponse	Received Response Data Area

Data Area	Description
D1-RetryDetails	Retry Details
D1-SOAPFaultDA	SOAP Fault
D1-SPMeterHistorySyncRequestIE	MDM SP/Meter History
D1-SPRelatedActivity	SP Related to Activity Data Area
D1-ScalToIntProfileFactor	Convert Scalar to Interval Profile Factors
D1-ScratchpadMCTypes	Scratchpad MC Types
D1-StandardDeviceEventType	Standard Device Event Type
D1-StandardLogFields	Standard Log Fields
D1-StatusCodeDescriptions	Status Code Descriptions Data
D1-SuccessResponse	Success Response Data Area
D1-SyncRequestInbound	Inbound Sync Request
D1-TraceData	Trace Data
D1-UsageCommon	Usage Common Data Area
D1-VEECommon	VEE Common Data Area
D1-VEEIMDSeeder	VEE IMD Seeder Initial Measurement Data

Oracle Utilities Meter Data Management Back Package Data Areas

The table below lists the available Oracle Utilities Meter Data Management base package data areas.

Data Area	Description
D2-CalcUsgException List	Calculate Usage Exception List
D2-CalcUsgScalarDetails	Calculate Usage Scalar Details
D2-CalcUsgSummaryUsgPeriod	Calculate Usage Summary Usage Period
D2-ChartCommon	Common Chart Data
D2-ChartEvents	Chart Events
D2-ChartSummaryInformation	Chart Summary Information
D2-CommonAggregator	Common Aggregator
D2-CommonDynamicOption	Common Dynamic Option Data Area
D2-CommonDynamicOptionEvent	Common Dynamic Option Event Data Area
D2-CommonDynamicOptionType	Common Dynamic Option Type Data Area
D2-CommonSyncRequestDetailsUS	Common Sync Request Details - Usage Transaction
D2-CommonUsageSubscription	Common Usage Subscription Data Area
D2-CommonUsageSubscriptionType	Common Usage Subscription Type Data Area
D2-DispatcherRuns	Save area for accumulated Dispatcher Runs
D2-EventBarProfiles	Event Bar Profiles
D2-FinalValuesGraphSummary	Final Values Graph Summary
D2-Functions	Functions
D2-GetScalarValList	Get Scalar Validation List
D2-Intervals	Vector Math Intervals
D2-MCChartCommon	Measuring Component Common Chart Fields
D2-MCFunctionCommon	Common Fields for Measuring Component Functions
D2-MsrmtQuantityAggregator	Measurement Quantity Aggregator
D2-SPChartCommon	Service Point-Based Common Chart fields
D2-TempArea	Area for sorting SQ list
D2-TempArea2	Area for sorting SQ list
D2-UTChartCommon	Usage-Based Common Chart Fields
D2-UsaagePeriodsLite	Usage Period Lite DA
D2-UsageTranExceptionList	Usage Transaction Exception List
D2-UsageTranIntervalMC	Usage Transaction Interval MC

Data Area	Description
D2-UsageTranScalarMC	Usage Transaction Scalar MC
D2-UsageTranTraceList	Usage Transaction Trace List
D2-UsageTransactionDetails	Usage Transaction Details
D2-VectorData	Vector Data
D2-VectorList	Vector List

Use the Data Area portal to view more details concerning these data areas.

Meter Data Framework Base Package Extendable Lookups

The table below lists the available meter data framework base package extendable lookups.

Business Object	Description
D1-360EventBarProfile	360 View Event Bar Profile
D1-DaysSinceLastNormalMsrmtLkp	Days Since Last Normal Measurement
D1-DeviceEventMappingLookup	Device Event Mapping
D1-DeviceLocationLookup	Device Location
D1-DvcCommunicationStatLookup	Device Communication Status
D1-DvcConnectionStatLookup	Device Connection Status
D1-DvcEventCategoryLookup	Device Event Category
D1-DvcFunctionalStateLookup	Device Functional State
D1-ExecutionPriorityLookup	Execution Priority
D1-ExternalActTypeIdentifier	External Activity Type Identifier
D1-ExternalCommTypeLookup	External Communication Type
D1-FinalValuesOverlayProfile	Final Values Overlay Profile
D1-HeadendUOMLookup	Headend UOM Code to Standard UOM
D1-KeyLookup	Key
D1-LogicalStatusLookup	Logical Status
D1-MeasurementConditionLookup	Measurement Condition
D1-OKToEnterLookup	Ok to Enter
D1-SPInstructionLookup	SP Instruction
D1-SPWarningLookup	SP Warning
D1-StdEventNameLookup	Standard Event Name
D1-TimeZoneTranslationLookup	Head-End Time Zone to Standard Mapping
D1-TransactionTypeLookup	Transaction Type

Use the Extendable Lookup portal to view more details concerning these extendable lookups.

Oracle Utilities Meter Data Management Back Package Extendable Lookups

The table below lists the available Oracle Utilities Meter Data Management base package extendable lookups.

Business Object	Description
D2-CCBRateScheduleLookup	CCB Rate Schedule
D2-ConsumptionSnapshotTypeLkup	Usage Snapshot Type
D2-DaysSinceLastUTLookup	Days Since Last Usage Transaction
D2-SPUTAgingSnapshotTypeLookup	Unreported Usage Analysis Snapshot Type

Use the Extendable Lookup portal to view more details concerning these extendable lookups.

Index

M

S