

**Oracle Utilities Smart Grid Gateway
Adapter for Sensus RNI**

Administrative User Guide

Release 2.2.0.2

E92029-01

January 2018

Oracle Utilities Smart Grid Gateway Adapter for Sensus RNI Administrative User Guide

Release 2.2.0.2

E92029-01

January 2018

Documentation build: 1.5.2018 15:59:11 [D6_1515185951000]

Copyright © 2011, 2018, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Sensus Adapter Overview.....	4
Sensus Adapter Processing.....	5
Initial Measurement Data and Device Event Loading.....	5
Initial Measurements.....	5
Device Events.....	8
Base Package Business Objects.....	10
Device Communication.....	10
Communication Flows.....	10
Device Communication Base Package Business Objects.....	11
External System.....	12
Outbound Message Types.....	12
Inbound / Outbound Service Configuration.....	12
BPEL Processes.....	14
Configuring a Sensus Head-End System.....	16
Inbound Web Services.....	16
Message Senders.....	17
Outbound Message Types.....	18
External System.....	18
Service Provider.....	18
Processing Methods.....	19
Configuring Endpoint URIs.....	20
Configuring Sensus Extendable Lookups.....	22
Extending the Sensus Adapter.....	24
The Sensus Test Harness.....	25
Test Harness Design.....	25
Locating the WSDL for the Test Harness.....	26
Web Services.....	26
General Services.....	26
Locate Meter Services.....	27
Meter Administration Services.....	29
Meter Attribute Administration Services.....	31

Chapter 1

Sensus Adapter Overview

The Oracle Utilities Smart Grid Gateway Adapter for Sensus RNI supports communication with the Sensus Regional Network Interface (RNI), including measurement data and device event loading, and command messaging in support of commissioning, connect, disconnect, decommissioning, status check, and on-demand read. The following table describes the attributes of the adapter:

Attribute	Details
Currently Supported Version	Sensus RNI 3.3
Protocol	MultiSpeak 3.0 & 4.1. RNI 3.1 and the SGG Sensus RNI adapter support MultiSpeak 4.1
Market(s)	Worldwide
Architecture	Long range radio WAN (mesh)

The adapter uses Oracle Service Bus (OSB) and Oracle Business Process Execution Language (BPEL) to facilitate communication between Oracle Utilities Smart Grid Gateway and the Sensus RNI.

The following functionality is included:

Measurement Data and Device Event Loading - data parsing and transformation via Oracle Service Bus from Sensus RNI format into the Oracle Utilities Service and Measurement Data Foundation unified format for measurement data and device events.

Measurement Data and Device Event Processing - configurable mapping for Sensus RNI status codes and device event names to Oracle Utilities Service and Measurement Data Foundation standard values.

Smart Meter Command Processing - sending/receiving messages to/from the Sensus RNI application to initiate smart meter commands from Oracle Utilities Smart Grid Gateway. The Sensus RNI adapter supports the following types of commands and communications:

- **Meter Commissioning** - business objects and BPEL processes to support issuing meter commissioning commands.
- **Meter Decommissioning** - business objects and BPEL processes to support issuing meter decommissioning commands.
- **Remote Connect** - business objects and BPEL processes to support issuing remote connect commands.
- **Remote Disconnect** - business objects and BPEL processes to support issuing remote disconnect commands
- **On-Demand Read** - business objects and BPEL processes to support issuing on-demand read commands.
- **Device Status Check** - business objects and BPEL processes to support issuing device status check commands.

Chapter 2

Sensus Adapter Processing

This section provides details concerning the OSB processing, BPEL Processes, and OUAF 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 customizations you create as part of your implementation.

Initial Measurement Data and Device Event Loading

The initial measurement data load and subsequent device event processing use OSB to poll for, parse, and transform the head-payloads into the Oracle Utilities Smart Grid Gateway service format. Payloads contain measurements and meter events in some head-end specific format OSB then places each service call into a JMS queue within the Oracle Utilities applications. The JMS client consumes the entries and invokes the respective services in parallel then a service creates initial measurements with data in a common format with head-end-specific processing as needed. A second service creates device events with data in a common format

Initial Measurements

The usage data exported from the AMI head-end system as a file in Sensus RNI format is loaded into Oracle Utilities as Initial measurement data. The following OSB projects, delivered in the base product, help manage the usage processing:

1. **SGG-D6-USAGE-BASE** contains components responsible for “actual” processing of incoming data. It should not be modified during configuration. This can be upgraded without affecting the customization and environment settings added to SGG-D6-USAGE-CM.
2. **SGG-D6-USAGE-CM** allows for customization and simplifies future upgrades.

The runtime configuration settings for the SGG-D6-USAGE-CM project are stored in the xquery file EnvironmentSettings.xq. You can use this file to adjust initial measurement data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRawIMD element.

The following table describes the elements included in the EnvironmentSettings.xq file:

Element	Description	Valid Values
populateRawIMD	Determines if the initial measurement data is populated as raw data.	true false
callPreProcessing	Determines if the preprocessing proxy service is called.	true false
callPostProcessing	Determines if the postprocessing proxy service is called.	true false
destinationRootElementInterval	Holds the name of inbound web service for the interval IMD seeder.	
destinationRootElementScalar	Holds the name of inbound web service for the scalar IMD seeder. In most cases it is the same as destinationRootElementInterval.	
dateTimelnUTC	Indicates whether the Sensus RNI system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device.	true false
publishServices/service	Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker).	
filterUsage	Determines if usage should be filtered.	true false
useExternalTOU	Indicates whether or not an externally reference TOU period should be used when processing measurement data.	true false

Publishing Initial Measurement Data

The Sensus adapter can be configured to publish initial measurement data for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

Enabling Initial Measurement Data Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>
  <service>[publisherBusinessService]</service>
</publishServices>
```

The following components provided with the SGG-D6-USAGE-CM OSB project are used in publishing measurement data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

Configuring Initial Measurement Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems.

Initial measurement data is published in the “native” initial measurement data format (the format of the initial measurement seeder business object). This format includes normalized unit of measure and condition codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

NOTE: Initial measurement data published via this feature is published prior to VEE processing. In addition, filtering can NOT be applied to data published via this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

Parameter	Description	Default Value
SGG_DR_INT_QUEUE	JNDI name of queue to publish SGG payloads. This is the JMS queue defined in the DataRakerBusinessService business service. This should NOT be changed.	DataRakerQueue
SOA_DR_PUBLISH_SIZE	The number of records (SGG payloads) to accumulate in a published file.	100
SOA_DR_FILE_SIZE	The maximum file size for the accumulated (SGG payloads) file in kilobytes.	524288
SOA_DR_ELAPSED_TIME	The period of time in second which, when exceeded, causes a new outgoing file to be created.	600
SOA_DR_POLLING_FREQ	The polling frequency in seconds of the staging directory for new files.	60
SOA_DR_STAGING_DIR	Mount point/directory for the staging directory for accumulated SGG payload files. This is used internally and should NOT be changed.	/spl/sploutput/staging
SOA_DR_INTEGRATION_DIR	Mount point/directory from which Oracle DataRaker will consume the converted XML files.	/spl/sploutput/int

Filtering Initial Measurement Data

The Sensus adapter can be configured to filter initial measurement data passed into Oracle Utilities Smart Grid Gateway and Meter Data Management. Filtering data is enabled by setting the <filterUsage> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterUsage>true</filterUsage>
```

When filtering is enabled, only measurements whose <externalUOM> matches one of the values defined in the **Sensus UOM Code to Standard UOM Mapping** extendable lookup (D6-HeadendUOMLookup) are passed into the system for processing.

Processing Large Input Files

In some environments, the OSB project may begin processing a large input file before it has been completely copied to the input directory. To prevent this, configure the MinimumAge property in the “InboundProxyService” proxy service for the SGG-D6-USAGE-CM project. The MinimumAge property specifies the minimum age of files to be retrieved, based on the last modified time stamp. This enables large files to be completely copied to the input directory before they are retrieved for processing.

Prioritized Initial Measurement Processing

The Sensus adapter prioritizes processing of initial measurements created from smart meter commands and/or completion events by setting the Execution Method flag in these types of initial measurements to “Real Time” (D1RT). In addition,

initial measurements received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing.

See **Initial Measurement Data Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

Device Events

The device event data exported from the head-end system as a file in Sensus RNI format is loaded into Oracle Utilities as a Device Event. One of your configuration tasks is to customize the device events processing.

The required functionality is delivered in the base product as two OSB projects:

1. **SGG-D6-EVENT-BASE** contains components responsible for “actual” processing of incoming data. It can be upgraded in future without affecting the customization and environment settings that done in SGG-D6-EVENT-CM project.
2. **SGG-D6-EVENT-CM** allows the customization and simplifies the future upgrades.

The runtime configuration settings for the SGG-D6-EVENT-CM project are stored in the xquery file EnvironmentSettings.xq. You can use this file to adjust initial measurement data processing. For example, if you want to load raw data you would specify “true” for the content of the populateRaw element.

The following table describes the elements included in the EnvironmentSettings.xq file:

Element	Description	Valid Values
populateRaw	Determines if the event data is populated as raw data.	true false
callPreProcessing	Determines if the preprocessing proxy service is called.	true false
callPostProcessing	Determines if the postprocessing proxy service is called.	true false
destinationRootElement	Holds the name of inbound web service for the device event seeder.	
dateTimelnUTC	Indicates whether the Sensus RNI system is sending date/time information in UTC (true) or local time of the device (false). If not provided the default behavior will be local time of the device.	true false
publishServices/service	Specifies the name of the business service within the OSB project used to publish data for external systems (such as Oracle DataRaker).	
filterEvents	Determines if events should be filtered.	true false

Publishing Events

The Sensus adapter can be configured to publish device events for use in Oracle DataRaker or other external systems. This functionality is supported through a combination of OSB components and BPEL composites.

Enabling Device Event Publishing

Publishing data is enabled by referencing a publisher business service in the publishServices/service element in the EnvironmentSettings.xq file as follows:

```
<publishServices>  
  <service>[publisherBusinessService]</service>  
</publishServices>
```


The following components provided with the SGG-D6-EVENT-CM OSB project are used in publishing device events data to Oracle DataRaker:

- The **DataRakerBusinessService** business service is used to send data to a pre-configured JMS queue (defined as an Endpoint URI), from which the data will be published. This is the business service that should be specified in the EnvironmentSettings.xq file.
- The **DataRakerServiceAccount** service account is used to define and maintain the user name and password needed to access the JMS queue defined in the **DataRakerBusinessService** business service.

Configuring Device Event Publishing Output

The SGGDRIntegration BPEL composite handles publishing the data to Oracle DataRaker or other systems.

Device event data is published in the “native” device event data format (the format of the device event seeder business object). This format includes normalized device event codes. See the *Oracle Utilities Smart Grid Gateway Adapter Development Kit Administrative User Guide* for more details about this format.

NOTE: In addition, filtering can NOT be applied to device events published via this feature.

The following parameters can be used to configure details of how the data is provided to Oracle DataRaker, including the directory where files are posted for Oracle DataRaker to consume, number of records per file, polling frequency, etc. These parameters are defined during installation. See the *Oracle Utilities Smart Grid Gateway Installation Guide* for more details about defining values for these parameters.

Parameter	Description	Default Value
SGG_DR_INT_QUEUE	JNDI name of queue to publish SGG payloads. This is the JMS queue defined in the DataRakerBusinessService business service. This should NOT be changed.	DataRakerQueue
SOA_DR_PUBLISH_SIZE	The number of records (SGG payloads) to accumulate in a published file.	100
SOA_DR_FILE_SIZE	The maximum file size for the accumulated (SGG payloads) file in kilobytes.	524288
SOA_DR_ELAPSED_TIME	The period of time in second which, when exceeded, causes a new outgoing file to be created.	600
SOA_DR_POLLING_FREQ	The polling frequency in seconds of the staging directory for new files.	60
SOA_DR_STAGING_DIR	Mount point/directory for the staging directory for accumulated SGG payload files. This is used internally and should NOT be changed.	/spl/sploutput/staging
SOA_DR_INTEGRATION_DIR	Mount point/directory from which Oracle DataRaker will consume the converted XML files.	/spl/sploutput/int

Filtering Events

The Sensus adapter can be configured to filter device events passed into Oracle Utilities Smart Grid Gateway and Meter Data Management. Filtering data is enabled by setting the <filterEvents> element in the EnvironmentSettings.xq file to “true” as follows:

```
<filterEvents>true</filterEvents>
```

When filtering is enabled, only device events whose <externalEventName> matches one of the values defined in the **Sensus Device Event Mapping** extendable lookup (D6-DeviceEventMappingLookup) are passed into the system for processing.

Prioritized Device Event Processing

The Sensus adapter prioritizes processing of device events created from smart meter commands and/or completion events by setting the Execution Method flag in these types of device events to “Real Time” (DIRT). In addition, device events received with the Execution Method flag set to “Real Time” will be processed in real time rather than via batch processing.

See **Device Event Prioritization** in the *Oracle Utilities Meter Data Management / Smart Grid Gateway Business User Guide* for more information.

Base Package Business Objects

The Sensus RNI adapter base package includes the following device and initial measurement business objects:

Business Object Name	Description
D6-InitialLoadIMDInterval	Sensus Initial Load IMD - Interval Used when loading Sensus interval measurements into the system for the first time.
D6-InitialLoadIMDScalar	Sensus Initial Load IMD - Scalar
D6-SmartMeter	Sensus Smart Meter

Device Communication

The basic communication for all business processing is essentially the same. A communication request is sent from the Oracle Utilities application to Sensus RNI. This request would be for a connect/disconnect, commission/decommission, measurement data, device status check, or an on-demand read. The designated BPEL process transforms the request from Oracle Utilities format to MultiSpeak format and invokes the related Sensus RNI web service. Sensus RNI then returns a reply, and the BPEL process transforms the reply message back to the appropriate format so that Oracle Utilities can receive the response.

Communication Flows

The table below lists the communications created for each Sensus RNI command:

Command	Outbound Communication	Inbound Communication	Completion Event
Remote Connect	Sensus – Initiate Connect Disconnect	Sensus – Connect / Disconnect State Change	Connect Device Completion Event
Remote Disconnect	Sensus – Initiate Connect Disconnect	Sensus – Connect / Disconnect State Change	Disconnect Device Completion Event
Device Commissioning	Sensus – Meter Add Notification		Device Commissioning Completion Event

Command	Outbound Communication	Inbound Communication	Completion Event
Device Decommissioning	Sensus – Meter Remote Notification		Device Decommissioning Completion Event
On-Demand Read (Scalar)	Sensus – Initiate Meter Read By Meter ID	Sensus – Reading Changed Notification	Create IMD Completion Event
On-Demand Read (Interval)	Interval data not supported		
Device Status Check	Sensus – Initiate Outage Detection	Sensus – Outage Detection Event Notification	

Device Communication Base Package Business Objects

The Sensus RNI Adapter base package includes the following communication business objects:

Business Object Name	Description
D6-ConnectDisconStateChgNtf	Sensus - Connect/Disconnect State Change
D6-InitiateConnectDisconnect	Sensus - Initiate Connect Disconnect
D6-InitiateMeterByMeterId	Sensus - Initiate Meter Read By Meter ID
D6-InitiateOutageDetection	Sensus - Initiate Outage Detection
D6-MeterAddNotification	Sensus - Meter Add Notification
D6-MeterRemoveNotification	Sensus - Meter Remove Notification
D6-OutageDetectEvtNotification	Sensus - Outage Detection Event Notification
D6-ReadingChgNotification	Sensus - Reading Changed Notification
D6-UnsolicitedEvtNotification	Sensus - Unsolicited Event Notification

Sensus Event Data Mapping

The Sensus event file format maps as follows into the business object, D1-DeviceEventMappingLookup:

Sensus Flat File Field	Device Event Seeder BO Element	Comments
Transaction ID (from Header record)	External Source Identifier	This is the file name.
Device Identifier	External Device Identifier	
Event Name	External Event Name	
Event Creation Date/Time	Event Date/Time	
Device Type	External Device Type	This element has no real bearing on the device type within MDM/SGG. Its valid values include (although the element itself is free-form): Meter Collector Router
Service Location ID	External Service Location ID	
Communication Module Serial Number	External Communication Module Identifier	
Event Category ID	External Event Category	
Event Severity	External Event Severity	Valid values include (although the element itself is free-form): Alert

Sensus Flat File Field	Device Event Seeder BO Element	Comments
		Information
Status Value	External Status Value	This represents additional information that relates to the event itself.
Status Date/Time	External Status Date/Time	The date & time at which the additional information referenced above had occurred.

External System

You must 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 or Real-time)
- The corresponding message senders
- Batch Control (if Processing Method is set to Batch)
- Message XSL, W3C Schema, and Response XSL (as applicable)

Outbound Message Types

Acknowledgement and response messages are sent and received validating that commands have been transmitted.

Outbound Message Type	Description
D6-CONDISCON	Sensus Initiate Connect Disconnect
D6-INITMTR	Initiate Meter Read By Meter ID
D6-INTOUTDET	Initiate Outage Detection Request

Inbound / Outbound Service Configuration

The inbound/outbound message utility allows you to configure your system to receive information from and to send information to external applications using XML. The Sensus RNI adapter for Smart Grid Gateway uses one inbound web service to map device events. This is the same inbound web service used by the D1 application.

Inbound Web Services

Inbound web 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 inbound web services you need to create is based on the types of messages the system is designed to send.

The Oracle Utilities Smart Grid Gateway adapter for Sensus RNI includes the following inbound web services:

Inbound Web Service	Description
D1-BulkRequestHeader	Bulk Request Header
D1-BulkRequestUpdate	Bulk Request Update
D1-BulkResponse	Bulk Response

Inbound Web Service	Description
D1-DeviceEventSeeder	<p>Used for upload of device events.</p> <p>The Device Event Seeder business object serves as a means of adding device events both from outside the application and from online. Its pre-processing algorithms determine the device event type - which in turn defines the device event BO that should be used to create the device event.</p> <p>If a device event type can't be determined, the device event is created using this BO. Such a device event can then be re-processed - and if successful, a new device event is created.</p>
D1-InitialLoadIMD	<p>Used for initial measurement upload.</p> <p>The IMDSeeder business object is used to determine the type of initial measurement business object to instantiate when receiving usage readings from a head-end system.</p>
D1-DeviceStatusCheck	<p>Device Status Check</p> <p>This service is invoked by the integration layer to instantiate a Device Status Check command.</p>
D1-InitialLoadIMD	<p>Used by OSB to instantiate an IMD</p> <p>This inbound web service is used by OSB to instantiate an Initial Measurement Data for incoming interval usage in the Sensus format.</p>
D1-RemoteConnect	<p>Remote Connect</p> <p>This service is invoked by the integration layer to instantiate a Remote Connect command.</p>
D1-RemoteDisconnect	<p>Remote Disconnect</p> <p>This service is invoked by the integration layer to instantiate a Remote Disconnect command.</p>
D6-ConDisconStChgNotification	<p>Initiate Connect Disconnect response.</p> <p>Retrieve response from the Initiate Connect Disconnect command.</p>
D6-OutageDetectionEventNotification	<p>Initiate Outage Detection Response</p> <p>Retrieve response from the Initiate Outage Detection Event Notification command.</p>
D6-ReadingChangedNotification	<p>Reading Changed Notification</p> <p>Notification that a Sensus device reading has changed.</p>
D6-UnsolicitedEventNotification	<p>Unsolicited Event Response</p> <p>Retrieve unsolicited notifications when an event triggers an alarm on the meter.</p>

Message Senders

Message 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 message senders you need to create is based on the types of messages the system is designed to accept.

The Oracle Utilities Smart Grid Gateway adapter for Sensus RNI includes the following message senders:

Message Sender	Description
D6-CONDISCON	Sensus Initiate Connect/Disconnect
D6-INTOUTDET	Initiate Outage Detection Request

Message Sender	Description
D6-InitMID	Initiate Meter Read By Meter ID Outbound Message

BPEL Processes

These processes are responsible for performing the conversion from Oracle Utilities format to MultiSpeak 4.1 format, invoking process callouts and invoking the remote endpoint to trigger the device events.

OnDemandRead Composite Process — Invokes the remote endpoint to trigger the on-demand read event. An asynchronous reply responds to the OUAF layer when the reading arrives.

ConnectDisconnect Composite Process — Invokes the remote endpoint to trigger the connect/disconnect event. An asynchronous reply responds to the OUAF layer when confirmation of the requested event arrives.

CommissionDecommission Composite Process — Invokes the remote endpoint to trigger the commission or decommission event. After the synchronous call completes, a one of the following second business callout services is invoked to determined if the related “received” or “completed” callout should be executed:

- isExecutingCommissionReceivedCallout
- isExecutingCommissionCompletedCallout
- isExecutingDecommissionReceivedCallout
- isExecutingDecommissionCompletedCallout

DeviceStatusCheck Composite — Invokes the remote endpoint to trigger the initiate outage detection event. An asynchronous reply responds to the OUAF layer when confirmation of the requested event arrives.

ProcessCallout Composite — This business callout provides a point at which customers and implementers can incorporate custom business logic and transformations. This composite includes the WSDLs and processing logic for all of the MultiSpeak processes. The default implementation of each method is a direct return of the input.

Web Services

These web services are all defined in the Sensus RNI head end system. The WSDLs were added to a Meta Data Storage (MDS) layer in OUAF and all references to the WSDL point to this MDS location. These web services have HTTP security by default. You may need to modify the security as a part of your implementation.

Web Service	Related BPEL Process	Description
CB_ServerService	ConnectDisconnect	<p>This web service defines the return interface, the means by which the status is returned to the calling system.</p> <p>This web service is only be invoked by the head end system, not OUAF. Only the CDStateChangeNotification web method is implemented in the composite.</p> <p>The endpointURI format is: http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Sensus/ConnectDisconnect/CB_ServerService</p>
CB_Server	OnDemandRead	<p>This web service defines the return interface, the means by which the reading is returned to the calling system.</p> <p>This web service is only be invoked by the head end system, not OUAF. Only the ReadingChangedNotification web method is implemented in the composite.</p>

Web Service	Related BPEL Process	Description
		The endpointURI format is: http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Sensus/OnDemandRead/CB_Server
OA_ServerService	DeviceStatusCheck	<p>This web service defines the asynchronous return for InitiateOutageDetectionEventRequest for solicited responses. It is also used for unsolicited alarms.</p> <p>This web service is only be invoked by the head end system, not OUAF. Only the ODEventNotification, PingURL, and GetMethods web methods are implemented in the composite.</p> <p>The endpointURI format is: http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Sensus/DeviceStatusCheck/OA_ServerService</p>

Sensus RNI Web Services

The following table describes the Sensus RNI web services and operations used for the Oracle Utilities Smart Grid Gateway command messaging:

Smart Grid Gateway Command	AMI Adapter Business Objects	Sensus Web Services	Sensus Operations
Device Commissioning	D6-MeterAddNotification	MR	MeterAddNotification
Device Decommissioning	D6-MeterRemoveNotification	MR	MeterRemoveNotification
Remote Connect/ Remote Disconnect	D6-InitiateConnectDisconnect	CD	InitiateConnectDisconnect
	D6-ConnectDisconStateChgNtf	CB	CDStatesChangedNotification (async reply)
Device Status Check	D6-InitiateOutageDetection	OD	InitiateOutageDetectionEventRequest
	D6-OutageDetectEvtNotification	OA	ODEventNotification (async reply)
On-Demand Read	D6-InitiateMeterByMeterId	MR	InitiateMeterReadingsByMeterID
	D6-ReadingChangedNotification	CB	ReadingChangedNotification (async reply)

Chapter 3

Configuring a Sensus Head-End System

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway Adapter for Sensus RNI to communicate with the Sensus RNI.

Inbound Web Services

Inbound web services define the details of how messages are received from an external system. This includes incoming usage and device events, as well as messages sent from the Sensus Regional Network Interface (RNI) in response to a command request.

The following inbound web services must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating inbound web services.

Inbound Web Service Name	Description
D1-DeviceEventSeeder	Device Event Seeder
D1-DeviceStatusCheck	Device Status Check
D1-InitialLoadIMD	IMD Seeder
D1-PayloadErrorNotif	Payload Error Notification
D1-PayloadStatistics	Payload Statistics
D1-PayloadSummary	Payload Summary
D6-ConDisconStChgNotification	Initiate Connect Disconnect Response
D6-OutageDetectionEventNotification	Initiate Outage Detection Response
D6-ReadingChangedNotification	Reading Changed Notification
D6-UnsolicitedEventNotification	Unsolicited Event Response

Note: The following apply to all of the above inbound web services:

Message Options

- **Trace:** No
- **Debug:** No

- **Active:** Yes

Operations

- **Operation Name:** Same as web service name
- **Schema Type:** Business Object
- **Schema Name:** Applicable business object code
- **Transaction Type:** Add

Message Senders

Message senders define the details of how messages are sent to an external system, such as messages containing device command requests.

The following message senders must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating message senders.

Message Sender	Description
D6-CONDISCON	Sensus Initiate Connect/Disconnect
D6-INTOUTDET	Initiate Outage Detection Request
D6-InitMID	Initiate Meter Read By Meter ID Outbound Message

Note: The following apply to all of the above message senders:

Main Tab:

- **Invocation Type:** Real-time
- **Message Class:** RTHHTTPSNDR (Sender routes message via HTTP real-time)
- **MSG Encoding:** UTF-8 message encoding

Context Tab:

- **HTTP Header:** SOAPAction:<OPERATION>
- **HTTP Login User:** <USER_ID>
- **HTTP Login Password:** <PASSWORD>
- **HTTP Method:** POST
- **HTTP URL 1:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/Sensus/<SERVICE>
- where:
 - <OPERATION>: the operation performed by the message sender (see Operation column in the table above)
 - <USER_ID>: the user ID used to log into WebLogic Enterprise Manager
 - <PASSWORD>: the password used to log into WebLogic Enterprise Manager
 - <EM_SERVER_IP>: the machine name or IP address of server where the WebLogic Enterprise Manager is installed
 - <EM_SERVER_PORT>: the port where the WebLogic Enterprise Manager is installed
 - <SERVICE>: the service invoked by the message sender (see Service column in the table above)

Outbound Message Types

Outbound message types define specific types of messages sent to an external system, such as messages containing device command requests.

The following outbound message types must be configured in your system. If these are not present in your configuration, add them. Refer to the Oracle Utilities Application Framework documentation for more information about creating outbound message types.

Outbound Message Type	Description
D6-CONDISCON	Sensus Initiate Connect Disconnect
D6-INITMTR	Initiate Meter Read By Meter ID
D6-INTOUTDET	Initiate Outage Detection Request

Note: The following apply to all of the above outbound message types:

- **Business Object:** D1-OutboundMessage (Outbound Message)
- **Priority:** Priority 50

External System

External systems represent external applications with which the Smart Grid Gateway will exchange messages or data. In the case of the Smart Grid Gateway adapters, external systems represent the head-end systems with which the adapters communicate.

An external system that represents the Sensus RNI must be present in your system. If this is not present in your configuration, add it, along with the following Outbound Message Types. Refer to the Oracle Utilities Application Framework documentation for more information about creating external systems.

External System - Sensus:

- **External System:** Sensus
- **Description:** Sensus
- **Outbound Message Types:**

Outbound Message Type	Description	Message Sender
D6-CONDISCON	Sensus Initiate Connect Disconnect	D6-CONDISCON
D6-INITMTR	Initiate Meter Read By Meter ID	D6-InitMID
D6-INTOUTDET	Initiate Outage Detection Request	D6-INTOUTDET

Note: The following apply to all of the above outbound message types:

- **Processing Method:** Real-time
- **Message XSL:** D6-Request.xml
- **Response XSL:** D6-Response.xml

Service Provider

Service providers represent external entities that serve various roles relative to the application, including 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. The head-end systems that collect and send measurement data and meter events to the application are defined as service providers.

A service provider that represents the Sensus RNI must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Meter Data Management / Smart Grid Gateway documentation for more information about creating service providers.

Service Provider - Sensus:

- **Service Provider:** Sensus
- **Description:** Sensus
- **External Reference ID:** Sensus
- **External System:** Sensus
- **Our Name/ID in Their System:**
- **AMI Device ID Type:** Internal Meter Number
- **AMI Measuring Component ID Type:** Channel ID

Processing Methods

Processing methods define the format or means by which a service provider receives and/or sends data from and/or to the application, including bill determinants, usage data, or device events. Processing methods are also used to define how to create information internal to the application such as initial measurement data and device events. Processing methods can also be used to define how command requests are sent to the Sensus RNI.

The following types of processing methods must be configured for the Sensus service provider. Refer to the Oracle Utilities Meter Data Management / Smart Grid Gateway documentation for more information about configuring processing methods.

Initial Measurement Creation

Initial measurement creation processing methods define the business objects used to create initial measurements. The IMD Seeder inbound web service uses this processing method to determine which type of initial measurement business object to instantiate when receiving usage from the Sensus RNI.

Device Event Mapping

Device event mapping processing methods define how head-end-specific device events are mapped to standard device event names. The Device Event Seeder inbound web service uses this processing method to determine which type of device event business object to instantiate when receiving device events from the Sensus RNI.

UOM Translation

UOM translation processing methods define how head-end-specific unit of measure (UOM) codes are translated to standard UOM codes. This processing method is used to determine how to map Sensus UOM codes to standard UOM codes when receiving usage from the Sensus RNI.

Commands

Command processing methods define how command requests are sent to a head-end system. More specifically, they define the type of outbound communication business object to create for each type of command, and the outbound message type to send to the head-end system.

The following types of command processing methods can be configured for the Sensus service provider, based on the requirements of each implementation using the “How to Create OB COMM/Send OB Message” processing method business object (D1–HowToCreateActivityOBComm).

Command	Processing Role	Default Business Object	Default Outbound Message Type
Device Commission	Device Registration	D6-MeterAddNotification	Commission
Device Decommission	Device Removal	D6-MeterRemoveNotification	Decommission
Device Status Check	Device Status Check	D6-InitiateOutageDetection	Initiate Outage Detection Request
On-Demand Read (Scalar)	On-Demand Read (Scalar)	D6-InitiateMeterByMeterId	Initiate Meter Read by Meter ID
Remote Connect	Remote Connect	D6-InitiateConnectDisconnect	Connect Device
Remote Disconnect	Remote Disconnect	D6-InitiateConnectDisconnect	Disconnect Device

Configuring Endpoint URIs

Part of configuring your Sensus adapter is configuring your BPEL composites to work with your head-end system by defining the appropriate Endpoint URIs for each of the commands.

The default approach to defining Endpoints URIs is redeployment or reinstallation of the BPEL composites. For example, by default, changing an adapter from using the test harness to a production environment using the actual head end system requires editing the appropriate installation menu options and redeploying the BPEL composites. The Endpoints URIs defined during installation and deployment for each adapter are listed in the **Smart Grid Gateway Installation and Configuration Worksheets** section of the *Oracle Utilities Smart Grid Gateway Installation Guide*.

You can also use an “Endpoint Override” Domain Value Map (DVM) to override Endpoints URIs defined during deployment and installation. This DVM allows defining specific keys that provide an alternate URL that will override the original installed value. Endpoint Override DVMs can be edited using the Oracle SOA Composer.

The Sensus endpoint override DVM (D6–EndpointOverrides.dvm) uses a specific set of keys, each used for one or more commands. The table below lists the DVM keys available for the Sensus adapter and the command used with each:

DVM Key	Commands
MR_Server	Device Commissioning Device Decommissioning On-Demand Read (Scalar)
CD_Server	Remote Connect Remote Disconnect
OD_Server	Device Status Check
ProcessCallout	User Exit Functions

To define an override Endpoint URI for the Sensus adapter, use the following procedure:

1. Open the SOA Composer for your BPEL configuration.
The URL for the SOA Composer is `http://server:port/soa/composer`.
2. Select the D6–EndpointOverrides.dvm in the left panel.
 - In Fusion Middleware v12.2.1, this is located under **Shared** in the **Deployment View**, or under **Domain Value Maps** in the **Types View**.
 - In Fusion Middleware v12.2.2, this is located under **Metadata** in the **Deployment View**.

The DVM will open in the right panel. The **Description** field lists the available keys for the DVM (only the first key is shown, but you can scroll through the contents to view the list. The panel also displays a list of previously defined keys.

3. Click the **Create Session** button (above the left panel) to begin an editing session.

4. Click the **Add Domain Values** icon (“+”) to add a new key.

The **Add Domain Values** dialog opens.

5. Enter the appropriate values in the **Add Domain Values** dialog as follows:

- **key**: The DVM key for the Endpoint URI you wish to define (see the table above).
- **EndpointURI**: The override Endpoint URI.

6. Click **OK**.

The new DVM value will appear in the list of keys.

7. Click the **Save** icon to save the DVM values.

8. Click the **Publish** button to activate all the changes in the editing session.

Enter an optional note for the session in the **Publish Session** dialog.

Changes take effect immediately upon publishing the session.

Click **Discard** to discard your changes.

Click **Exit** to exit your current session. Note that your session will still be open if you exit. Use **Discard** to end your session without making changes.

Chapter 4

Configuring Sensus Extendable Lookups

This section outlines some of the extendable lookups that must be configured for use with the Sensus adapter. Refer to the Oracle Utilities Application Framework documentation for more information about working with extendable lookups.

Sensus Device Event Mapping

The Sensus Device Event Mapping extendable lookup is used to determine which type of device event business object to instantiate when receiving device events from the Sensus RNI.

Each value defined for the Sensus Device Event Mapping extendable lookup should include the following:

- **Head-End System Event Name:** The event name used by the Sensus RNI
- **Description:** A description of the device event
- **Status:** The status of the lookup value (can be Active or Inactive)
- **Standard Event Name:** The standard event name for device events of this type, from the “Standard Event Name” extendable lookup.

Sensus UOM Code to Standard UOM Mapping

Usage received from Sensus may use utility-specific unit of measures (UOMs). These custom UOMs must be mapped to standard UOM codes. The Sensus UOM Code to Standard UOM Mapping extendable lookup is used to determine how to map Sensus UOM codes to standard UOM codes when receiving usage from the Sensus RNI.

Each value defined for the Sensus UOM Code to Standard UOM Mapping extendable lookup should include the following:

- **Unit of Measure:** The unit of measure defined in the system.
- **Head-end UOM:** The unit of measure code used by the Sensus RNI
- **Description:** A description of the unit of measure code.
- **Status:** The status of the lookup value (can be Active or Inactive)

Sensus Interval Status Code to Condition Mapping

Interval usage received from the Sensus RNI can include Sensus interval status codes that indicate the status or condition of the interval value. These interval status codes must be mapped to standard condition codes in the system. The Sensus

Interval Status Code to Condition Mapping extendable lookup is used to determine how to map Sensus interval status codes to standard status codes when receiving usage from the Sensus RNI.

Each value defined for the Sensus Interval Status Code to Condition Mapping extendable lookup should include the following:

- **Interval Status:** The Sensus interval status code
- **Description:** A description of the interval status code.
- **Status:** The status of the lookup value (can be Active or Inactive)
- **Condition:** The condition code to which the interval status code is to be mapped, from the Measurement Condition extendable lookup.

Other Extendable Lookups

Business Object Name	Description
D6-CDReasonCodeLookup	Sensus Connect/Disconnect Reason Code
D6-LoadActionCodeLookup	Sensus Load Action Code
D6-OutageEventTypeLookup	Sensus Outage Event Type
D6-SensusTimeUnits	Sensus Time Units
D6-ServiceTypeMappingLookup	Sensus Service Type Mapping

Chapter 5

Extending the Sensus Adapter

The Oracle Utilities Smart Grid Gateway Adapter for Sensus supports a number of commands, including:

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

The Adapter for Sensus can be extended to support additional commands provided by the Sensus RNI.

Chapter 6

The Sensus Test Harness

Oracle Utilities Smart Grid Gateway Adapter for Sensus RNI includes a test harness that can be configured to simulate a general head-end system for testing the two-way commands. The test harness includes a BPEL composite, web services for standard meter functions, and an XML file that can be used to contain information for one or more meters. This chapter describes the test harness and its components.

Test Harness Design

The Sensus RNI Harness is divided into two main layers. The “front end” set of services implements the Sensus RNI specified interfaces. They receive requests corresponding to the following:

- http://www.multispeak.org/Version_4.1_Release MR_Server
 - MeterAddNotification
 - MeterRemoveNotification
 - InitiateMeterReadingsByMeterID
- http://www.multispeak.org/Version_4.1_Release CD_Server
 - InitiateConnectDisconnect
- http://www.multispeak.org/Version_4.1_Release CB_Server
 - CDStatesChangedNotification (async reply)
 - ReadingChangedNotification (async reply)
- http://www.multispeak.org/Version_4.1_Release OD_Server
 - InitiateOutageDetectionEventRequest

Each of these services calls into the “back end” layer, which defines meters and sets their attributes. These meters are stored in a file within the test harness called meterdb.xml. This file can be modified pre-deployment. Post-deployment changes to the file are not supported. However, the Test Harness retains an in-memory “database” of the meters in the file. The in-memory representation can be modified using the Utility web services. Note that any changes to the in-memory structure will be lost when the server is restarted or the Test Harness composite is redeployed.

Test Harness Security Credentials

The test harness is configured with the following default Multispeak credentials for RNI security:

- MultiSpeakUserID/MultiSpeakPwd

For more information about configuring security for the SOA system for the Sensus RNI adapter, see the *Oracle Utilities Smart Grid Gateway Installation Guide*.

Locating the WSDL for the Test Harness

Follow these procedures to locate the test harness WSDL:

How to Use Enterprise Manager to Locate the WSDL

1. Open Enterprise Manager and use the navigation pane to open the dashboard of the test harness composite:
2. The top bar of the dashboard contains several buttons and icons. One of these is a “world” icon with a puzzle piece over it. Click this icon to display a list of the WSDLs and endpoint URIs for the composite:
3. Click the UtilService WSDL URL link to see the WSDL in the browser, or right click and save it to your machine

Depending on your requirements, it may be necessary to download the associated schema found in the wsdl:types section. The URL can be pasted into a browser tab and downloaded in the same manner as the WSDL. The main schema has imported schemas that may also be required.

How to Use a Direct URL to locate the WSDL

The WSDL can be accessed without Enterprise Manager by understanding the paths used on the SOA server. In general, they have the following form:

```
http://{server name}:{port number}/soa-infra/services/{partition}/{Composite}/{Web Service}?WSDL
```

So by default, the test harness WSDL can be found at

```
http://{server name}:{port number}/soa-infra/services/Sensus_Test/Sensus/UtilService?WSDL
```

Web Services

This section describes the web services included in the Sensus test harness BPEL composite.

General Services

This section describes the general services of the Sensus test harness composite.

LoadMeterIndex

This web service loads the data store from the internal file. By default if the store is already in memory, it will *not* reload. This behavior can be overridden with the forceReload parameter.

Input: LoadMeterIndexInput

Part: payload

Element: LoadMeterIndexRequest

Parameter	Description
forceReload	A switch telling the system whether to reload the meter index from the configuration file. Default value is false.

Output: LoadMeterIndexOutput

Part: payload

Element: LoadMeterIndexResult

Parameter	Description
loaded	A boolean value for whether or not the index was reloaded from the configuration file

Fault: UtilityFault (see [UtilityFault](#) for more details).

ViewAuditTrail

This web service returns the audit log for the entire session.

Input: ViewAuditTrailInput

Part: payload

Element: ViewAuditTrailRequest

Parameters: This is an empty request. There are no parameters.

Output: ViewAuditTrailOutput

Part: payload

Element: ViewAuditTrailResult

This element is an entry consisting of a timestamp and an Operation. Each entry may have an associated meter object showing what changed.

Fault: UtilityFault (see [UtilityFault](#) for more details).

UtilityFault

Fault with similar mapping to SGG/OUAF faults:

Typically, the faultCode, faultString, faultActor, and detail/text elements will be populated.

Locate Meter Services

This section describes the locate meter web services of the Sensus test harness composite.

FindMeters

This web service queries the data store for one or more meters. The difference between GetMeter and FindMeters is GetMeter can return at most one meter and it must match the provided ID exactly. GetMeter will throw an error if the ID is not found. FindMeters can return more than one meter (when using the regex) and will not throw an error when the ID does not match any of the meters in the index.

Input: FindMetersInput

Part: payload

Element: FindMetersRequest

Parameter	Description
id	The meter ID for which to search
isRegex	The provided id can be a regex value when this parameter is true. Hint: to search for all meters in the system, use ".*" for the ID.

Output: FindMetersOutput

Part: payload

Element: FindMetersResult

Zero or more meter objects can be returned from the search

Fault: See [UtilityFault](#). Unlike other methods, FindMeters does not throw an exception if the meter is not found. As such, it can be used to test for the existence of a Meter prior to querying for it.

IsMeterDefined

This web service queries whether a particular meter is defined in the data store.

Input: IsMeterDefinedInput

Part: payload

Element: IsMeterDefinedRequest

Parameter	Description
id	The meter ID for which to search

Output: IsMeterDefinedOutput

Part: payload

Element: IsMeterDefinedResult

Whether or not the provided ID is part of the index.

Fault: See [UtilityFault](#). Thrown when meter id is not found.

GetMeter

This web service returns all the attributes of a single meter from the in-memory data store. The difference between GetMeter and FindMeters is GetMeter can return at most one meter and it must match the provided ID exactly. GetMeter will throw an error if the ID is not found. FindMeters can return more than one meter (when using the regex) and will not throw an error when the ID does not match any of the meters in the index.

Input: GetMeterInput

Part: payload

Element: GetMeterRequest

Parameter	Description
id	The meter ID for which to search

Output: GetMeterOutput

Part: payload

Element: GetMeterResult

The meter object requested by the ID.

Fault: See [UtilityFault](#). Thrown when meter id is not found.

Meter Administration Services

This section describes the meter administration services of the Sensus test harness composite.

AddMeters

This web service adds a set of meters to the in-memory data store. This will not permanently add it to the control file.

Input: AddMetersInput

Part: payload

Element: AddMetersRequest

Parameter	Description
id	The identification code for the meter.
utility	An informational string.
serviceType	One of the valid ServiceType values (see schema). "Electric" is the only option at this time.
isCommissioned	Whether or not the meter is in a commissioned state.
loadActionCode	One of the possible LoadActionCode values used in Connect and Disconnect (see schema).
outageEventType	One of the possible OutageEventType values used in Device Status Check (see schema).
executionStatus	One of the possible ExecutionStates (see schema). These values control how the meter will respond to commands.
updateIfExisting	Whether or not to update the meter with the provided values if it already exists in the index.
Comment	An informational string describing the purpose of the meter.
Channels	A listing of unit of measures supported by this meter.
uomCode	A code describing the unit of measure for the channel.
uomName	A short string containing the name of the unit of measure.
decimals	The number of digits to the right of the decimal that should be generated when reading the meter.
description	A longer description of the unit of measure.

Output: AddMetersOutput

Part: payload

Element: AddMetersResult

Whether or not each meter was added to the index.

Fault: See [UtilityFault](#).

RemoveMeter

This web service removes a meter from the in-memory data store. This will not permanently remove it from the control file.

Input: RemoveMeterInput

Part: payload

Element: RemoveMeterRequest

Parameter	Description
id	The ID for the meter to be removed.

Output: RemoveMeterOutput

Part: payload

Element: RemoveMeterResult

Whether or not the meter was removed from the index.

Fault: See [UtilityFault](#). Thrown when meter id is not found.

AddMeterChannel

This web service adds a new channel to a single meter.

Input: AddMeterChannelInput

Part: payload

Element: AddMeterChannelRequest

Parameter	Description
id	The identification code for the meter.
uomCode	A code describing the unit of measure for the channel.
uomName	A short string containing the name of the unit of measure.
decimals	The number of digits to the right of the decimal that should be generated when reading the meter.
description	A longer description of the unit of measure.

Output: AddMeterChannelOutput

Part: payload

Element: AddMeterChannelResult

Whether or not the channel was added to the index.

Fault: See [UtilityFault](#). Thrown when meter id is not found.

RemoveMeterChannel

This web service removes a Channel from a meter.

Input: RemoveMeterChannelInput

Part: payload

Element: RemoveMeterChannelRequest

Parameter	Description
id	The ID for the meter to be removed.
uomCode	A code describing the unit of measure for the channel.
uomName	A short string containing the name of the unit of measure.

These three parameters are combined to locate a unique channel

Output: RemoveMeterChannelOutput

Part: payload

Element: RemoveMeterChannelResult

Whether or not the channel was removed from the meter.

Fault: See [UtilityFault](#). Thrown when meter id is not found.

ReadScalarMeter

This web service generates a scalar reading for each channel of a given meter.

Input: ReadScalarMeterInput

Part: payload

Element: ReadScalarMeterRequest

Parameter	Description
id	The ID for the meter to be read.

Output: ReadScalarMeterOutput

Part: payload

Element: ReadScalarMeterResult

Zero or more scalar readings for the given meter.

Parameter	Description
uomCode	A code describing the unit of measure for the channel.
uomName	A short string containing the name of the unit of measure.
decimals	The number of digits to the right of the decimal that should be generated when reading the meter.
description	A longer description of the unit of measure.
value	A random number representing the scalar reading.

Fault: See [UtilityFault](#). Thrown when meter id is not found.

Meter Attribute Administration Services

This section describes the meter administration services of the Sensus test harness composite.

GetOutageEventType

This web service queries the outage event type for a given meter. The OutageEventType is used by DeviceStatusCheck.

Input: GetOutageEventTypeInput

Part: payload

Element: GetOutageEventTypeRequest

Parameter	Description
id	The ID for the meter for which the OutageEventType should be retrieved.

Output: GetOutageEventTypeOutput

Part: payload

Element: GetOutageEventTypeResult

The value of the OutageEventType attribute for the requested meter.

Fault: See [UtilityFault](#). Thrown when meter id is not found.

SetOutageEventType

This web service updates the outage event type for a given meter.

Input: SetOutageEventTypeInput

Part: payload

Element: SetOutageEventTypeRequest

Parameter	Description
id	The ID for the meter for which the OutageEventType should be set.
value	The new value of OutageEventType to set on the meter.

Output: SetOutageEventTypeOutput

Part: payload

Element: SetOutageEventTypeResult

The boolean response indicates the success or failure of the update (not the current field status).

Fault: See [UtilityFault](#). Thrown when meter id is not found.

GetLoadActionCode

This web service queries the load action code for a given meter. This is the Connect/Disconnect behavior.

Input: GetLoadActionCodeInput

Part: payload

Element: GetLoadActionCodeRequest

Parameter	Description
id	The ID for the meter for which the LoadActionCode should be retrieved.

Output: GetLoadActionCodeOutput

Part: payload

Element: GetLoadActionCodeResult

The value of the LoadActionCode attribute for the requested meter.

Fault: See [UtilityFault](#). Thrown when meter id is not found.

SetLoadActionCode

This web service updates the load action code for a given meter.

Input: SetLoadActionCodeInput

Part: payload

Element: SetLoadActionCodeRequest

Parameter	Description
id	The ID for the meter for which the LoadActionCode should be set.
value	The new value of LoadActionCode to set on the meter.

Output: SetLoadActionCodeOutput

Part: payload

Element: SetLoadActionCodeResult

The boolean response indicates the success or failure of the update (not the current field status).

Fault: See [UtilityFault](#). Thrown when meter id is not found.

IsCommissioned

This web service queries the commissioning status for a given meter. This is the Commission/Decommission behavior.

Input: IsCommissionedInput

Part: payload

Element: IsCommissionedRequest

Parameter	Description
id	The ID for the meter for which the Commissioned status should be retrieved.

Output: IsCommissionedOutput

Part: payload

Element: IsCommissionedResult

The value of the Commissioned status attribute for the requested meter.

Fault: See [UtilityFault](#). Thrown when meter id is not found.

SetCommissioned

This web service updates the commissioning status for a given meter.

Input: SetCommissionedInput

Part: payload

Element: SetCommissionedRequest

Parameter	Description
id	The ID for the meter for which the Commissioned status should be set.
value	The new value of Commissioned status to set on the meter.

Output: SetCommissionedOutput

Part: payload

Element: SetCommissionedResult

The boolean response indicates the success or failure of the update (not the current field status).

Fault: See [UtilityFault](#). Thrown when meter id is not found.

GetExecutionStatus

This web service queries the status of the property controlling the overall execution of the command.

Input: GetExecutionStatusInput

Part: payload

Element: GetExecutionStatusRequest

Parameter	Description
id	The ID for the meter for which the ExecutionStatus should be retrieved.

Output: GetExecutionStatusOutput

Part: payload

Element: GetExecutionStatusResult

The value of the ExecutionStatus attribute for the requested meter.

Fault: See [UtilityFault](#). Thrown when meter id is not found.

SetExecutionStatus

This web service updates the property controlling the overall completion of the command.

Input: SetExecutionStatusInput

Part: payload

Element: SetExecutionStatusRequest

Parameter	Description
id	The ID for the meter for which the ExecutionStatus should be set.
value	The new value of ExecutionStatus to set on the meter.

Output: SetExecutionStatusOutput

Part: payload

Element: SetExecutionStatusResult

The boolean response indicates the success or failure of the update (not the current field status).

Fault: See [UtilityFault](#). Thrown when meter id is not found.