Oracle Utilities Smart Grid Gateway Adapter for Silver Spring Networks

Administrative User Guide

Release 2.2.0.1

E80272-02

April 2017

ORACLE®

Oracle Utilities Smart Grid Gateway Adapter for Silver Spring Networks Administrative User Guide

Release 2.2.0.1

E80272-02

April 2017

Documentation build: 4.10.2017 11:49:5 [D7_1491850145000]

Contents

Chapter 1

Silver Spring Networks UIQ Adapter Overview

The Oracle Utilities Smart Grid Gateway Adapter for Silver Spring Networks supports communication with the Silver Spring Networks UtilityIQ application, 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 Versions	UtilityIQ Version 4.10
Protocol	Proprietary
Market(s)	Worldwide
Architecture	RF WAN (mesh) based on Access Points

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 Silver Spring Networks UtilityIQ application.

The following functionality is included:

Measurement Data and Device Event Loading - data parsing and transformation via Oracle Service Bus from Silver Spring Networks 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 Silver Spring Networks 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 Silver Spring Networks application to initiate smart meter commands from Oracle Utilities Smart Grid Gateway. The Silver Spring Networks adapter supports the following types of commands and communications:

- **Meter Commissioning -** business objects and BPEL processes to support issuing meter commissioning 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
- **Meter Decommissioning** - business objects and BPEL processes to support issuing meter decommissioning 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

Silver Spring Networks 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 Silver Spring Networks XML 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-D7-SSNXML-BASE contains components responsible for "actual" processing of incoming data. It should not be modified during configuration. This project can be upgraded without affecting the customization and environment settings added to SGG-D7-SSNXML-CM.

2. SGG-D7-SSNXML-CM allows for customization and simplifies future upgrades.

The runtime configuration settings for the SGG-D7-SSNXML-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
populateRaw	Determines if the initial measurement data is populated as raw data. Valid values are:	true false
callPreProcessing	Determines if the preprocessing proxy service is called. Valid values are:	true false
callPostProcessing	Determines if the postprocessing proxy service is called. Valid values are:	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.	
destinationRootElementEvent	Holds the name of inbound web service for the device event seeder.	
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
filterUsage	Determines if usage should be filtered.	true false

Publishing Initial Measurement Data

The Silver Spring Networks adapter can be configured to publish initial measurement data for use in Oracle DataRaker or other external systems. 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>
```

Publishing measurement data to Oracle DataRaker is supported by the following components provided with the SGG-D7-SSNXML-CM OSB project:

- The **DataRakerBusinessService** business service is used to publish data to a specified JMS queue (defined as an Endpoint URI), from which the external system can receive the data. 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.

Initial measurement data is published in the "native" initial measurement data format (the format of the initial measurement seeder business object).

Filtering Initial Measurement Data

The Silver Spring Networks 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 **SSN - UOM Code to Standard UOM Mapping** extendable lookup (D7-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-D7-SSNXML-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 Silver Spring Networks 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 Silver Spring Networks CSV 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-D7-CSV-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-D7-CSV-CM project.

2. SGG-D7-CSV-CM allows the customization and simplifies the future upgrades.

The runtime configuration settings for the SGG-D7-CSV-CM project are stored in the EnvironmentSettings.xq XQuery file. You can use this file to adjust device event 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	Valie Values
populateRaw	Determines if the device event data is populated as raw data. Valid values are:	true false
callPreProcessing	Determines if the preprocessing proxy service is called. Valid values are:	true false
callPostProcessing	Determines if the postprocessing proxy service is called. Valid values are:	true false
destinationRootElement	Holds the name of inbound web service for the device event seeder.	
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. Valid values are:	true false

Publishing Events

The Silver Spring Networks adapter can be configured to publish device events for use in Oracle DataRaker or other external systems. 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>
```

Publishing device events to Oracle DataRaker is supported by the following components provided with the SGG-D7-CSV-CM OSB project:

• The **DataRakerBusinessService** business service is used to publish data to a specified JMS queue (defined as an Endpoint URI), from which the external system can receive the data. 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.

Device event data is published in the "native" device event data format (the format of the device event seeder business object).

Filtering Events

The Silver Spring Networks 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 **SSN - Device Event Mapping** extendable lookup (D7-DeviceEventMappingLookup) are passed into the system for processing.

Prioritized Device Event Processing

The Silver Spring Networks 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" (D1RT). 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 Silver Spring Networks adapter base package includes the following device and initial measurement business objects:

Business Object Name	Description
D7-InitialLoadIMDInterval	SSN - Initial Load IMD - Interval
	Used when loading Silver Spring Network (SSN) interval measurements into the system for the first time.
D7-InitialLoadIMDScalar	SSN - Initial Load IMD - Scalar
	Used when loading Silver Spring Network (SSN) scalar measurements into the system for the first time.
D7-SmartMeter	SSN — 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 Silver Spring Networks. 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 Silver Spring Networks format and invokes the related Silver Spring Networks web service. Silver Spring Networks 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 Silver Spring Networks command:

Command	Outbound Communication	Inbound Communication	Completion Event
Remote Connect	SSN–Connect or Disconnect	SSN – Connect or Disconnect Response	Connect Device Completion Event
		Remote Provisioning Job (Get Status)	
Remote Disconnect	SSN–Connect or Disconnect	SSN – Connect or Disconnect Response	Disconnect Device Completion Event
Device Commissioning	SSN- Replace Location		Device Commissioning Completion Event
Device Decommissioning	SSN- Replace Device At Location (Decomm)		Device Decommissioning Completion Event
On-Demand Read (Scalar)	SSN – Add Meter Read Job (Scalar)	SSN – Meter Read Response (Scalar)	Create IMD Completion Event
On-Demand Read (Interval)	SSN – Add Meter Read Job (Interval)	SSN – Meter Read Response (Interval)	Create IMD Completion Event
Device Status Check	SSN – Add Ping job	SSN – Ping Job Response	

Device Communication Base Package Business Objects

The Silver Spring Networks Adapter base package includes the following communication business objects:

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

Silver Spring Networks Event Data Mapping

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

Silver Spring 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 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

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

Outbound Message Type	Description
D7-COMMS	Replace Device At Location
D7-OB MSG TY	Outbound Message Type SSN

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 Silver Spring Networks 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 Silver Spring Networks includes the following inbound web services:

Inbound Web Service	Description
D1-BulkRequestHeader	Bulk Request Header
D1-BulkRequestUpdate	Bulk Request Update
D1-BulkResponse	Bulk Response
D1-DeviceEventSeeder	Used for upload of device events.
	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.
	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.
D1-DeviceStatusCheck	Device Status Check
	This service is invoked by the integration layer to instantiate a Device Status Check command.
D1-InitialLoadIMD	Used by OSB to instantiate an IMD
	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.
D1-RemoteConnect	Remote Connect
	This service is invoked by the integration layer to instantiate a Remote Connect command.
D1-RemoteDisconnect	Remote Disconnect
	This service is invoked by the integration layer to instantiate a Remote Disconnect command.
D7-ConnectDisconnectResponse	Connect Disconnect Response
	Retrieves response for Remote provisioning Job Connect or Disconnect commands.

Inbound Web Service	Description
D7-GetStatusResponse	D7-GetStatusResponse
	Retrieve response from the Get Status command.
D7-MeterReadResponseInterval	SSN - Meter Read Response (Interval)
D7-MeterReadResponseScalar	SSN - Meter Read Response (Scalar)
D7-PingJobResponse	SSN - Ping Response
	Retrieves response from the Ping Job Response command.

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.

BPEL Processes

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

OnDemandRead Composite Process — Provides access points to edge application and handles data between edge application and head end system. It invokes sequence of web methods to head end system and retrieves meter read and send it back to Edge application.

ConnectDisconnect Composite Process — Performs the conversion from Oracle Utilities format to SSN format, invokes process callouts, and invokes the remote endpoint to trigger the connect event. A second, asynchronous reply will call back into the OUAF layer when the status change is completed at the head-end system. Another asynchronous reply will call back into the OUAF layer to send Meter Read Results.

CommissionDecommission Composite Process — Performs the conversion from Oracle Utilities format to SSN UIQ format, invokes process callouts, and invokes the remote endpoint to trigger the commission or decommission of meter.

DeviceStatusCheck Composite — Performs the conversion from Oracle Utilities format to SSN format, invokes process callouts, and makes a call via a proxy to the head-end system starting the Meter Ping operation. In an ideal scenario, the job status is returned as completed and the results are acquired and sent back to OUAF. If the job takes longer, OUAF will initiate a second request that will poll the head end system for the job status. When the job is completed, the results are returned to OUAF.

Common Composite — Contains two main classes of operations: Proxies and ProcessCallouts. Proxies are simple mediators that forward a web service call to a preset endpoint. No transformations are performed. They are convenient because they allow head end URLs and security to be set in a single composite. ProcessCallouts are points of customization which allow users to modify data and/or initiate some external business process.

BulkRequest Composite — Provides access points to requesting application. It decouples the bulk request into single commands for each meter/device in the request and sends it to edge application for processing.

Web Services

The following web services are all defined in the Silver Spring Networks head-end system:

- **CommissionDecommissionService**
  - **BPEL Process:** CommissionDecommission
  - **Operation:** ReplaceDeviceAtLocation

- **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/CommissionDecommission/ CommissionDecommissionService
- **CommissionDecommissionService**
  - **BPEL Process:** CommissionDecommission
  - **Operation:** ReplaceLocation
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/CommissionDecommission/ CommissionDecommissionService
- **ConnectDisconnectService**
  - **BPEL Process:** ConnectDisconnect
  - **Operation:** AddRemoteProvisioningJob
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/ConnectDisconnect/ ConnectDisconnectService
- **DeviceStatusCheckService**
  - **BPEL Process:** DeviceStatusCheck
  - **Operation:** AddPingJob
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/DeviceStatusCheck/ DeviceStatusCheckService
- **AddMeterReadJobService**
  - **BPEL Process:** AddMeterRead
  - **Operation:** AddMeterReadJob
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/OnDemandRead/ AddMeterReadJobService
- **OnDemandReadService**
  - **BPEL Process:** OnDemandRead
  - **Operation:** GetJobStatus
  - **Enpoint URL:** http://<EM_SERVER>:<EM_SERVER_PORT>/soa-infra/services/SSN/OnDemandRead/ OnDemandReadService

Silver Spring Networks Utility IQ Web Services

The following table describes the Silver Spring Networks Utility IQ web services and operations used for the Oracle Utilities Smart Grid Gateway command messaging:

Smart Grid Gateway Command	AMI Adapter Business Objects	Silver Spring Networks Web Services	Silver Spring Networks Operations
Device Commissioning	D7-ReplaceLocation	Device Manager	findDevice
			ReplaceLocation
Device Decommissioning	D7-ReplaceDeviceAtLocForDecomm	Device Manager	findDevice
			ReplaceDeviceAtLocation
Remote Connect/ Remote Disconnect	D7-ConnectDisconnect	Device Manager	findDevice
	D7-GetStatus	Job Manager	addRemoteProvisioningJob
		DeviceResults	getJobStatusForDevice
			getRemoteProvisioningResultsByJobID

Smart Grid Gateway Command	AMI Adapter Business Objects	Silver Spring Networks Web Services	Silver Spring Networks Operations
			findJob
			getJobStatus
			getMeterReadResultsByJobID
Device Status Check	D7-AddPingJob	Device Manager	findDevice
		Job Manager	addPingJob, getJobStatus
		DeviceResults	getPingResultsByJobID
On-Demand Read	D7-AddMeterReadJobInterval	Device Manager	findDevice
	D7-AddMeterReadJobScalar	Job Manager	addMeterReadJob
		DeviceResults	getJobStatus
			getMeterReadResultsByJobID

Chapter 3

Configuring a Silver Spring Networks Head-End System

This section outlines the configuration required for the Oracle Utilities Smart Grid Gateway Adapter for Silver Spring Networks to communicate with the Silver Spring Networks UtilityIQ.

Master Configurations

Master Configurations are sources of global parameter records used by a system implementation. This section describes the master configuration that is specific to Oracle Utilities Smart Grid Gateway Adapter for Silver Spring Networks. Refer to the Oracle Utilities Meter Data Management / Smart Grid Gateway documentation For more information about other master configurations used by Oracle Utilities Smart Grid Gateway.

SSN Version Master Configuration

This master configuration specifies the version of the Silver Spring Networks head-end system with which the system is communicating. The configuration options are SSN Version 4.4 and SSN Version 4.7. Only one version can be specified at a time. Because of SSN version compatibility, configuration option 4.4 can be used to communicate with SSN Version 4.4 or 4.5, and configuration option 4.7 can be used to communicate with SSN Versions 4.6, 4.7, or 4.8.

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 Silver Spring Networks UtilityIQ application 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-BulkRequestHeader	Bulk Request Header
D1-BulkRequestUpdate	Bulk Request Update
D1-BulkResponse	Bulk Response
D1-DeviceEventSeeder	Device Event Seeder
D1-DeviceStatusCheck	Device Status Check
D1-InitialLoadIMD	Used by OSB to instantiate an IMD
D1-PayloadErrorNotif	Payload Error Notification
D1-PayloadStatistics	Payload Statistics
D1-PayloadSummary	Payload Summary
D7-ConnectDisconnectResponse	Connect Disconnect Response
D7-GetStatusResponse	Get Status Response
D7-MeterReadResponseInterval	SSN - Meter Read Response (Interval)
D7-MeterReadResponseScalar	SSN - Meter Read Response (Scalar)
D7-PingJobResponse	SSN - Ping 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. An message sender should be configured for each command.

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
D7-DECOMM	SSN Replace Device at Location for Decommission
D7-COMM	SSN Replace Location - Commission
D7-ADDJOB	SSN Add Meter Read Job
D7-ADDPING	SSN Add Ping Job
D7-CONNECT	SSN Connect Device
D7-GTSTATUS	SSN Get Status

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

Main Tab:

- **Invocation Type**: Real-time

- **Message Class**: RTHTTPSNDR (Sender routes message via HTTP real-time)
- **MSG Encoding**: UTF-8 message encoding

Context Tab:

- **HTTP Header**: SOAPAction:http://xmlns.oracle.com/ouaf/ssn/**<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/SSN/**<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
D7-COMMS	Replace Device At Location
D7-OB MSG TY	Outbound Message Type SSN

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 Silver Spring Networks UtilityIQ 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 - Silver Spring Networks:

- **External System**: Silver Spring Networks

- **Description**: Silver Spring Networks

- **Outbound Message Types**:

Outbound Message Type	Description
D7-OB MSG TY	Outbound Message Type SSN

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

- **Processing Method**: Real-time
- **Message XSL**: D7-Request xsl
- **Response XSL**: D7-Response.xsl

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 Silver Spring Networks UtilityIQ must be present in your system. If this is not present in your configuration, add it. Refer to the Oracle Utilities Service and Measurement Data Foundation documentation for more information about creating service providers.

Service Provider - Silver Spring Networks:

- **Service Provider**: Silver Spring Networks
- **Description**: Silver Spring Networks
- **External Reference ID**: Silver Spring Networks
- **External System**: Silver Spring Networks
- **Our Name/ID in Their System**:
- **AMI Device ID Type**: Internal Meter Number
- **Pre-Commissioning Device ID Type**:
- **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 Silver Spring Networks UtilityIQ.

The following types of processing methods must be configured for the Silver Spring Networks service provider. Refer to the Oracle Utilities Service and Measurement Data Foundation 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 Silver Spring Networks UtilityIQ.

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 Silver Spring Networks UtilityIQ.

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 Silver Spring Networks 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	D7-ReplaceLocation	Commission
Device Decommission	Device Removal	D7-ReplaceDeviceAtLocForDecomm	Decommission
Device Status Check	Device Status Check	D7-AddPingJob	Get Status
On-Demand Read (Scalar)	On-Demand Read (Scalar)	D7-AddMeterReadJobScalar	Read Meter Data
On-Demand Read (Interval)	On-Demand Read (Interval)	D7-AddMeterReadJobInterval	Read Meter Data
Remote Connect	Remote Connect	D7-ConnectDisconnect	Connect Device
Remote Disconnect	Remote Disconnect	D7-ConnectDisconnect	Disconnect Device

Configuring Endpoint URIs

Part of configuring your Silver Springs Network 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 or 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 Silver Springs Network endpoint override DVM (D7–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 Silver Springs Network adapter and the command used with each:

DVM Key	Commands
DeviceResults4.4	Remote Connect
DeviceResults4.7	Remote Disconnect
DeviceResults4.10	On-Demand Read (Scalar)
	On-Demand Read (Interval)
	Device Status Check
DataAggregation4.4	Not used in this version.
DataAggregation4.7	
DataAggregation4.10	
DeviceManager4.4	Device Commissioning
DeviceManager4.7	Device Decommissioning
DeviceManager4.10	Remote Connect
	Remote Disconnect
	On-Demand Read (Scalar)
	On-Demand Read (Interval)
	Device Status Check
JobManager4.4	Remote Connect
JobManager4.7	Remote Disconnect
JobManager4.10	On-Demand Read (Scalar)
	On-Demand Read (Interval)
	Device Status Check
ProcessCallout	User Exit Functions

NOTE:

The numbers in the keys above (4.4, 4.7, and 4.10) designate the version of the Silver Springs Network IQ head-end system.

Only a single version of each key should be defined, based on the version of the Silver Springs Network IQ head-end system

To define an override Endpoint URI for the Silver Springs Network 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 D7–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 Silver Spring Networks Extendable Lookups

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

Silver Spring Networks Device Event Mapping

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

Each value defined for the Silver Spring Networks Device Event Mapping extendable lookup should include the following:

- **Head-End System Event Name**: The event name used by the Silver Spring Networks UtilityIQ

- **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.

Silver Spring Networks UOM Code to Standard UOM Mapping

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

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

- **Head-End Unit of Measure**: The unit of measure code used by the Silver Spring Networks UtilityIQ

- **Description**: A description of the unit of measure code.

- **Status**: The status of the lookup value (can be Active or Inactive)

- **Unit of Measure**: The unit of measure defined in the system.

Silver Spring Networks Interval Status Code to Condition Mapping

Interval usage received from the Silver Spring Networks UtilityIQ can include Silver Spring Networks 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 Silver Spring Networks Interval Status Code to Condition Mapping extendable lookup is used to determine how to map Silver Spring Networks interval status codes to standard status codes when receiving usage from the Silver Spring Networks UtilityIQ.

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

- **Interval Status**: The Silver Spring Networks 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
D7-ActivationStatusLookup	SSN - Activation Status
D7-ChannelStatusLookup	SSN - Channel Status
D7-DataTimeZoneLookup	SSN - Data Time Zone
D7-DeviceStatusLookup	SSN - Device Status
D7-ExecutionStatusLookup	SSN - Execution Status
D7-HeadendSQILookup	SSN - Head-End SQI to Standard SQI
D7-HeadendTOULookup	SSN - Head-End TOU to Standard TOU
D7-IntervalStatusLookup	SSN - Interval Status
D7-MeterConnectionStatusLookup	SSN - Meter Connection Status
D7-MeterReadTypeLookup	SSN - Meter Read Type
D7-OldAdminStateNameLookup	SSN - Old Admin State Name
D7-PriorityLookup	SSN - Priority Status
D7-ProvisioningActionLookup	SSN - Provisioning Action
D7-ProvisioningCommandStatus	SSN - Provisioning Command Status
D7-ProvisioningDataType	SSN - Provisioning Data Type
D7-ReadModeLookup	SSN - Read Mode
D7-RegisterReadSourceLookup	SSN - Register Read Source
D7-RegisterStatusLookup	SSN - Register Status
D7-SSNVersionLookup	SSN - Version
D7-StatusFlagNameLookup	SSN - Status Flag Name

Chapter 5

Extending the Silver Spring Networks Adapter

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

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

The Adapter for Silver Spring Networks can be extended to support additional commands provided by the Silver Spring Networks UtilityIQ.

Chapter 6

The Silver Spring Networks Test Harness

Oracle Utilities Smart Grid Gateway Adapter for Silver Spring Networks (SSN) includes a test harness that can be configured to simulate the Silver Spring Networks UtilityIQ 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 SSN Harness is divided into two main layers. A "front end" set of services implements the SSN-specified interfaces in both the 4.4 version and in the 4.7 version. They receive requests corresponding to:

- urn:com:ssn:schema:service:v1.4:DataAggregation and urn:com:ssn:schema:service:v1.6:DataAggregation
  - getMeterFieldStatus
- urn:com:ssn:schema:service:v1.4:DeviceManager and urn:com:ssn:schema:service:v1.6:DeviceManager
  - FindDevice
  - ReplaceDeviceAtLocation
  - ReplaceLocation
- urn:com:ssn:schema:service:v1.4:DeviceResults and urn:com:ssn:schema:service:v1.6:DeviceResults
  - getRemoteProvisioningResultsByJobID
  - getMeterReadResultsByJobID
  - getPingResultsByJobID
- urn:com:ssn:schema:service:v1.4:JobManager and urn:com:ssn:schema:service:v1.6:JobManager
  - addRemoteProvisioningJob
  - getJobStatus
  - addMeterReadJob
  - addPingJob

- findJob
- getJobStatusForDevice

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.

Locating the WSDL for the Test Harness

Follow these procedures to locate the Silver Spring Networks 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/SSN_Test/SSNTestHarness/UtilService?WSDL
```

Web Services

This section describes the web services included in the Silver Spring Networks test harness BPEL composite

General Services

This section describes the general services of the Silver Spring Networks 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 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

Output: ViewAuditTrailOutput

Part: payload

Element: ViewAuditTrailResult

An Entry consisting of a timestamp and an Operation. Each entry may have an associated meter object showing the latest update.

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 Silver Spring Networks 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

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.

IsMACIDDefined

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

Input: IsMACIDDefinedInput

Part: payload

Element: IsMACIDDefinedRequest

Parameter	Description
id	The MAC address for which to search

Output: IsMACIDDefinedOutput

Part: payload

Element: IsMACIDDefinedResult

Whether or not the provided MAC address 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.

GetMeterByMACID

This web service returns all the attributes of a single meter from the in-memory data store. The difference between GetMeter and GetMeterByMACID is this method looks for a MAC address rather than the identifier of the meter. GetMeterByMACID will throw an error if the MAC address is not found.

Input: GetMeterByMACID

Part: payload

Element: GetMeterByMACIDRequest

Parameter	Description
id	The MAC address for which to search

Output: GetMeterByMACIDOutput

Part: payload

Element: GetMeterByMACIDResult

The meter object requested by the MAC address.

Fault: See UtilityFault. Thrown when meter id is not found.

GetGroupMeters

This web service retrieves the set of meters with the specified group name

Input: GetGroupMetersInput

Part: payload

Element: GetGroupMetersRequest

Parameter	Description
id	The group name common to the meters to retrieved

Output: GetGroupMetersOutput

Part: payload

Element: GetGroupMetersResult

The set of meters with the provided group name.

Fault: See UtilityFault. Thrown only in unusual circumstances.

Meter Administration Services

This section describes the meter administration services of the Silver Spring Networks 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.
macID	A MAC address that must be unique within the system.
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.
groupName	The name linking multiple meters together into a set.
jobExecutionStatus	One of the possible Job Execution Status values (see schema). This attribute determines how requested jobs perform.
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	Element defining one or more units of measure supported by this meter.
tier	Attribute that specifies the number of scalar readings returned by the meter. (Default = 1).
Channel	Element that defines a single channel recorded by 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.

Parameter	Description
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

Element: tier

Indicates the specific tier for each reading returned by the test harness. Each tier can contain zero or more scalar readings for the given meter, containing the following.

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.

Meter Attribute Administration Services

This section describes the meter administration services of the Silver Spring Networks test harness composite.

GetMACID

This web service queries the MAC address of a meter. MAC addresses are used to identify meters within the SSN system. Therefore, they must be unique within the test harness. The structure of a MAC address is six sets of character pairs separated by colons. A "real" MAC address has further limitations on the range of the data, but for the purposes of the test harness, any digit or character from A-Z will be returned.

Input: GetMACIDInput

Part: payload

Element: GetMACIDRequest

Parameter	Description
id	The ID for the meter for which the MAC address should be retrieved

Output: GetMACIDOutput

Part: payload

Element: GetMACIDResult

The MAC address for the requested meter.

Fault: See UtilityFault. Thrown when meter id is not found.

SetMACID

This web service updates the MAC address for a given meter. MAC addresses are used to identify meters within the SSN system. Therefore, they must be unique within the test harness. The structure of a MAC address is six sets of character pairs separated by colons. A "real" MAC address has further limitations on the range of the data, but for the purposes of the test harness, any digit or character from A-Z will be returned.

Input: SetMACIDInput

Part: payload

Element: SetMACIDRequest

Parameter	Description
id	The ID for the meter for which the MAC address should be set
value	The new MAC address to set on the meter

Output: SetMACIDOutput

Part: payload

Element: SetMACIDResult

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.

GetJobExecutionStatus

This web service queries the job execution status of a meter. Many activities in SSN initiate jobs which can end in one of several statuses. The job execution status controls the ending status. The possible values of a jobExecutionStatus are:

Status	Description
JOB_EXEC_STATUS_NOT_STARTED	The job has not started. This is the initial state of submitted jobs.
JOB_EXEC_STATUS_RUNNING	The job is running.
JOB_EXEC_STATUS_COMPLETE	The job has completed.
JOB_EXEC_STATUS_FAILURE	The job failed.
JOB_EXEC_STATUS_UNKNOWN	The job's status is unknown.

Other - The job's status is not one of the other values.

The enumeration values come from the SSN schema type "ExecutionStatusEnumeration."

Input: GetJobExecutionStatusInput

Part: payload

Element: GetJobExecutionStatusRequest

Parameter	Description
id	The ID for the meter for which the job execution status should be retrieved

Output: GetJobExecutionStatusOutput

Part: payload

Element: GetJobExecutionStatusResult

The value of the job execution status for the requested meter.

Fault: See UtilityFault. Thrown when meter id is not found.

SetJobExecutionStatus

This web service update the job execution status for a given meter. Many activities in SSN initiate jobs which can end in one of several statuses. The job execution status controls the ending status. The possible values of a jobExecutionStatus are:

Status	Description
JOB_EXEC_STATUS_NOT_STARTED	The job has not started. This is the initial state of submitted jobs.
JOB_EXEC_STATUS_RUNNING	The job is running.
JOB_EXEC_STATUS_COMPLETE	The job has completed.
JOB_EXEC_STATUS_FAILURE	The job failed.
JOB_EXEC_STATUS_UNKNOWN	The job's status is unknown.

Other - The job's status is not one of the other values.

The enumeration values come from the SSN schema type "ExecutionStatusEnumeration."

Input: SetJobExecutionStatusInput

Part: payload

Element: SetJobExecutionStatusRequest

Parameter	Description
id	The ID for the meter for which the job execution status should be set
value	The new job execution status to set on the meter

Output: SetJobExecutionStatusOutput

Part: payload

Element: SetJobExecutionStatusResult

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.

GetDeviceStatus

This web service queries the device status of a meter. The OutageEventType is used by DeviceStatusCheck. The possible values of a deviceStatusType are:

Status	Description
Instantaneous	The meter responds immediately to the status check
SlowResponse	An EndpointFailure with a FailureReason of "ErrorResponse" will be returned from the Test Harness
NoResponse	An EndpointFailure with a FailureReason of "NoResponse" will be returned from the Test Harness
Unknown	An EndpointFailure with a FailureReason of "Unspecified" will be returned from the Test Harness

Input: GetDeviceStatusInput

Part: payload

Element: GetDeviceStatusRequest

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

Output: GetDeviceStatusOutput

Part: payload

Element: GetDeviceStatusResult

The value of the device status for the requested meter.

Fault: See UtilityFault. Thrown when meter id is not found.

SetDeviceStatus

This web service updates the device status for a given meter. The possible values of a deviceStatusType are:

Status	Description
Instantaneous	The meter responds immediately to the status check

Status	Description
SlowResponse	An EndpointFailure with a FailureReason of "ErrorResponse" will be returned from the Test Harness
NoResponse	An EndpointFailure with a FailureReason of "NoResponse" will be returned from the Test Harness
Unknown	An EndpointFailure with a FailureReason of "Unspecified" will be returned from the Test Harness

Input: SetDeviceStatusInput

Part: payload

Element: SetDeviceStatusRequest

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

Output: SetDeviceStatusOutput

Part: payload

Element: SetDeviceStatusResult

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.

GetConnectionStatus

This web service queries whether the given meter is connected or disconnected. This method is used by the Connect/Disconnect service. The values for connectionStatusType are:

- Connected
- Disconnected
- Unknown

Input: GetConnectionStatusInput

Part: payload

Element: GetConnectionStatusRequest

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

Output: GetConnectionStatusOutput

Part: payload

Element: GetConnectionStatusResult

The connection status of the requested meter.

Fault: See UtilityFault. Thrown when meter id is not found.

SetConnectionStatus

This web service updates the load action code for a given meter. This method is used by the Connect/Disconnect service. The values for connectionStatusType are:

- Connected
- Disconnected
- Unknown

Input: SetConnectionStatusInput

Part: payload

Element: SetConnectionStatusRequest

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

Output: SetConnectionStatusOutput

Part: payload

Element: SetConnectionStatusResult

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 service is used by the Commission/ Decommission process. The commissioning attribute can be true or false.

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.

SetCommission

This web service updates the commissioning status for a given meter. This service is used by the Commission/ Decommission process. The commissioning attribute can be true or false.

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. The possible values of execution status are:

Status	Description
Success	The command should complete successfully
ResponseTimeout	The asynchronous response will never arrive
SyncOperationFail	A simulated fault will occur in the during the initial request
AsyncOperationFailure	A simulated fault will occur in the asynchronous response

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. The possible values of execution status are:

Status	Description
Success	The command should complete successfully
ResponseTimeout	The asynchronous response will never arrive
SyncOperationFail	A simulated fault will occur in the during the initial request

Status	Description
AsyncOperationFailure	A simulated fault will occur in the asynchronous response

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.