Oracle® Utilities Integration for Device Operations

Implementation Guide

Oracle Utilities Meter Data Management v2.0.1.8
Oracle Utilities Operational Device Management v2.0.1

Release 11.1

E36210-01

October 2012

ORACLE

Oracle® Utilities Integration for Device Operations Implementation Guide, Release 11.1

E36210-01

Primary Author: Oracle Corporation

Contents

Preface

Audience

This document is intended for anyone implementing the Oracle® Utilities Integration for Device Operations.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support
Oracle customers have access to electronic support through My Oracle Support. For information, visit:
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info

or

http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, refer to the following documents:

- Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack 11g Release 1 (11.1.1.6.0)
- Oracle Utilities Meter Data Management User's Guide
- Oracle Utilities Operational Device Management User's Guide

Additional Resources

The additional documentation provided specific to this release includes:

- Release Notes
- Installation Guide
- Implementation Guide

Part 1

About the Integration

This section provides an overview of the participating applications and information regarding the business processes addressed by this integration.

This section contains the following chapters:

Chapter 1

Overview

This document provides configuration and administration information for the integration between Oracle Utilities Operational Device Management (ODM) and Oracle Utilities Meter Data Management (MDM)

1.1 Additional Resources

The following additional resources are available:

Additional Resources and Location

Resources	Location
Oracle Utilities Integration for Device Operations Installation Guide	Same folder as this document, with the distribution for this product.
Oracle Utilities Meter Data Management Documentation for Release v2.0.1.8	Refer to Oracle Utilities Meter Data Management documentation located on the Oracle Software Delivery Cloud. https://edelivery.oracle.com/
Oracle Utilities Operational Device Management Documentation for Release v2.0.1	Refer to Oracle Utilities Operational Device Management documentation located on the Oracle Software Delivery Cloud. https://edelivery.oracle.com/

Note: Latest version of these documents are available on the Oracle Technology Network at http://www.oracle.com/technetwork/index.html

1.2 About the Integration Product

This section provides general information about the functionality and processing of Oracle Utilities Integration for Device Operations. This is an Application Integration Architecture (AIA) Direct Integration using the Service-Oriented Architecture (SOA) Suite and does not require the AIA Foundation Pack to be installed.

1.2.1 Oracle Utilities Operational Device Management

Oracle Utilities Operational Device Management provides functionality to mange the receipt, installation, maintenance, tracking and removal of large volumes of assets

1.2.2 Oracle Utilities Meter Data Management

Oracle Utilities Meter Data Management supports the loading, Validation, Editing, and Estimation (VEE) of meter data - from meter configuration, to meter read and usage validation to bill determinant calculations.

1.3 Supported Business Processes

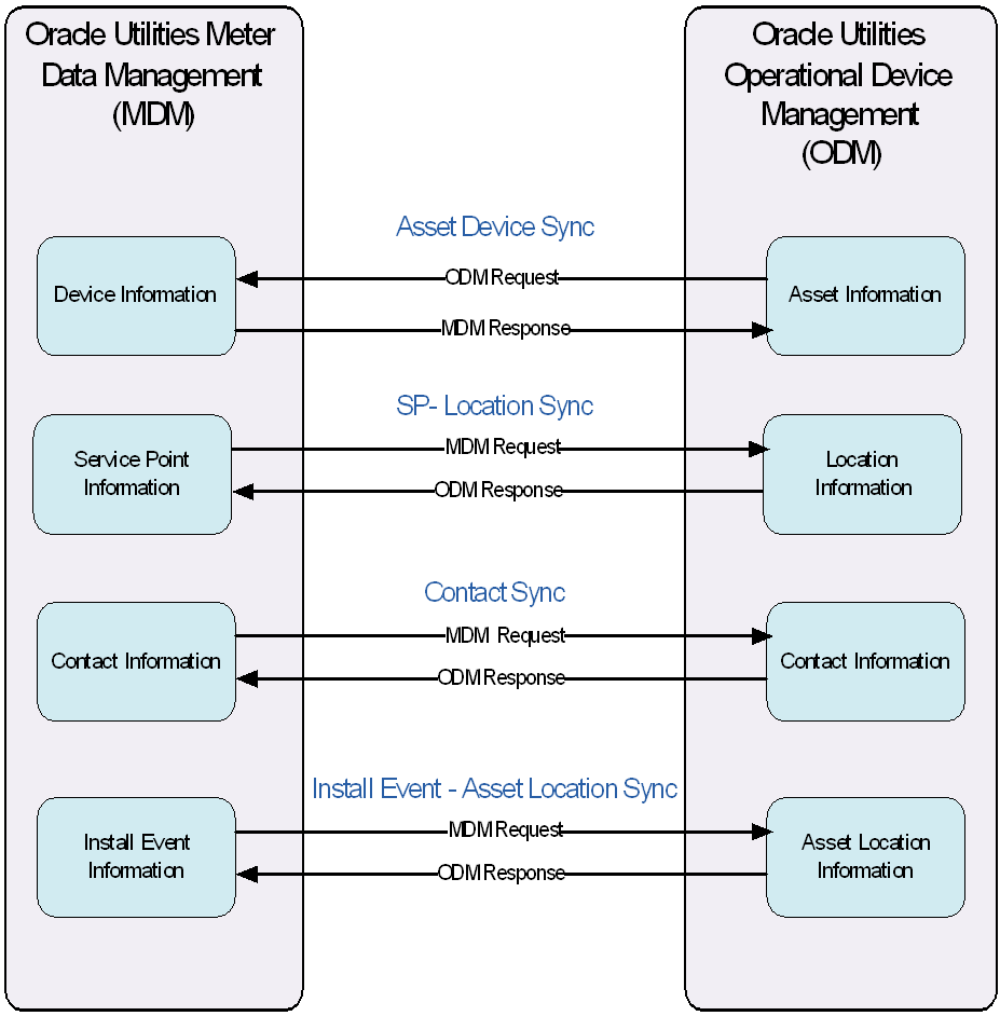
In this integration, Oracle Utilities Operational Device Management manages Assets (Devices).

The business processes are as follows:

- Asset-Device Synchronization
- SP - Location Synchronization
- Contact Synchronization
- Install Event – Asset Location Synchronization

Note that only the Asset-Device synchronization flow is initiated from Oracle Utilities Operational Device Management.

The following diagram provides a visual representation of this processing:



Integration Points

Chapter 2

Understanding the Integration Processes

This section outlines the overall technical overview, business processes, and specific integration points handled by the integration.

- Technical Overview
- Integration Points

2.1 Technical Overview

This section discusses the technicalities involved in this integration. They are as follows:

- This is a direct integration between Oracle Utilities Operational Device Management and Oracle Utilities Meter Data Management.
- All the end-to-end integration flows are asynchronous.
- The integration will send messages to JMS queues and receive messages from JMS queues. Oracle Utilities Operational Device Management and Oracle Utilities Meter Data Management will have the ability to read messages from JMS queues, and then write the processed messages to JMS queues.
- WebLogic JMS is used as a queuing mechanism in the integration layer. For each integration flow there are 8 JMS queues.

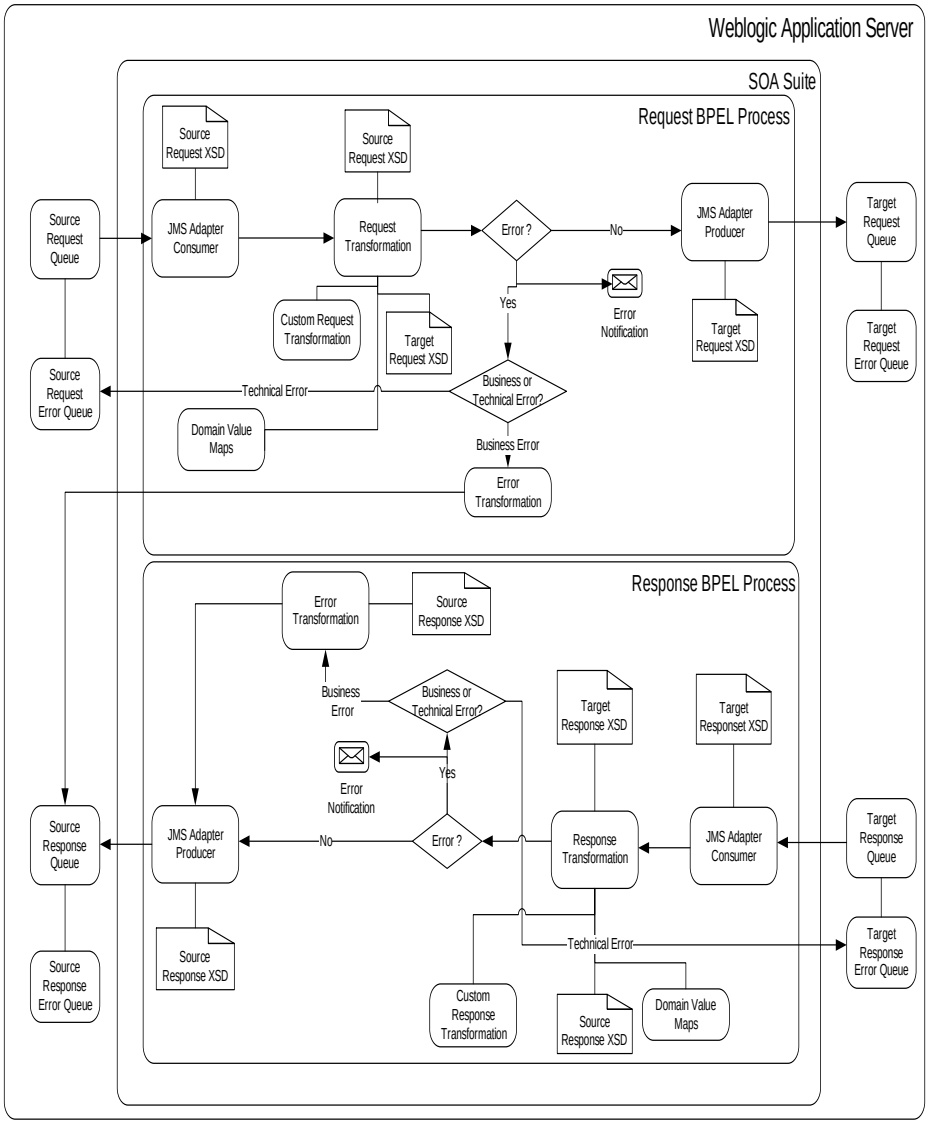
Technical Overview Details

S.No	Queue	Description
1	Source application request queue	Source application adds messages to this queue, which is then picked up by the integration layer for transformation.
2	Source application response queue	Business errors in the integration and success or failure acknowledgements from the target application are transformed by the integration and written to this queue.
3	Source application request error queue	Technical errors in the integration request flow are written to this queue.
4	Source application response error queue	Technical errors when the source application reads the messages from the source response queue are written to this queue.
5	Target application request queue	Integration reads messages from the source application request queue, transforms them, and writes them to this queue.
6	Target application response queue	Success or failure acknowledgements from the target application are written to this queue and are read by the integration.
7	Target application request error queue	Technical errors written when the target application tries to read the message from the target request queue are written to this queue.
8	Target application response error queue	Technical errors in the integration response flow are written to this queue.

- Two Business Process Execution Language (BPEL) processes manage each integration flow; one for the request processing, and one for the response processing.
- The Request BPEL process includes the following:
  - JMS Consumer to read from source request queue
  - JMS Producer to write to the target request queue
  - Transformations to convert messages from source format to target format. Domain Value Maps (DVMs) are used for the transformation.
  - Error handling and error notification

- The Response BPEL process includes the following:
  - JMS Producer to read from the target response queue
  - JMS Producer to write to the source response queue
  - Acknowledgement transformations to convert messages from the target format to the source format. Domain Value Maps (DVMs) are used for the transformation.
  - Error handling and error notification
- The JMS consumer and BPEL process is configured to participate in a global transaction, so that BPEL process can issue rollback and commits on the queue. The BPEL process issues rollbacks on the queue in the scenario where it is not able to reach the target queue and the message is moved to the corresponding error queue.
- All technical errors in the integration layer are moved to the error queue of the queue from which the message has been consumed.

The following diagram provides a graphical representation of this processing:



Technical Flow Diagram

2.2 Integration Points

The integration scope will support the following business processes:

- Data synchronization of new devices from Oracle Utilities Operational Device Management to Oracle Utilities Meter Data Management
- Data synchronization of service points and associated contacts from Oracle Utilities Meter Data Management to Oracle Utilities Operational Device Management
- Data synchronization of installs/removes from Oracle Utilities Meter Data Management to Oracle Utilities Operational Device Management

The key integration points for this integration, which will be an AIA direct integration, are as follows:

- Asset-Device Synchronization
- SP - Location Synchronization
- Contact Synchronization
- Install Event – Asset Location Synchronization

  Note: Location information is stored as Node information in Oracle Utilities Operational Device Management. Asset Location information is stored as Asset Node information in Oracle Utilities Operational Device Management.

2.2.1 Master Data Synchronization Processes

The following Integration points are available in Oracle Utilities Integration for Device Operations.

- Meter and component data are synchronized from Oracle Utilities Operational Device Management to Oracle Utilities Meter Data Management System.
- Contact, Service Point, and Install Event data are synchronized from Oracle Utilities Meter Data Management to Oracle Utilities Operational Device Management
- The integration uses the sync request process to capture data changes and communicate or synchronize the data changes between the two systems by sending out a sync request message to the integration layer.
- The sync request process is implemented using business object technology; therefore, much of the business rules and processing logic are defined in the algorithms associated to the business objects used by the sync process.
- The Audit plug-in spot defined on the entity's MO is used to detect changes in Oracle Utilities Meter Data Management data and to create sync requests. The Audit plug-in spot on the entity's BO are used to detect changes in Oracle Utilities Operational Device Management data and to create sync requests. These two plug-ins work in the same way, except Oracle Utilities Operational Device Management tracks changes at a BO level. Since only a subset of Oracle Utilities Operational Device Management assets (i.e., meters and components) are synced to Oracle Utilities Meter Data Management, having the audit plug-in at the BO level avoids the unnecessary creation of sync requests that would have to be eventually discarded.

- The sync request life cycle captures the change in data, sends sync request message to the integration, and awaits an acknowledgement back from the external system - whether positive or negative. Timeouts and negative acknowledgements received from the external system results in the sync request being transitioned to the Error state. As an option, implementation may choose to create a To Do entry in this case.
- There is a portal used for searching and viewing sync requests.

For more information about the sync request process, the business objects, maintenance objects, and other components used for this process, see the Oracle Utilities Framework User Guide section titled "Data Synchronization."

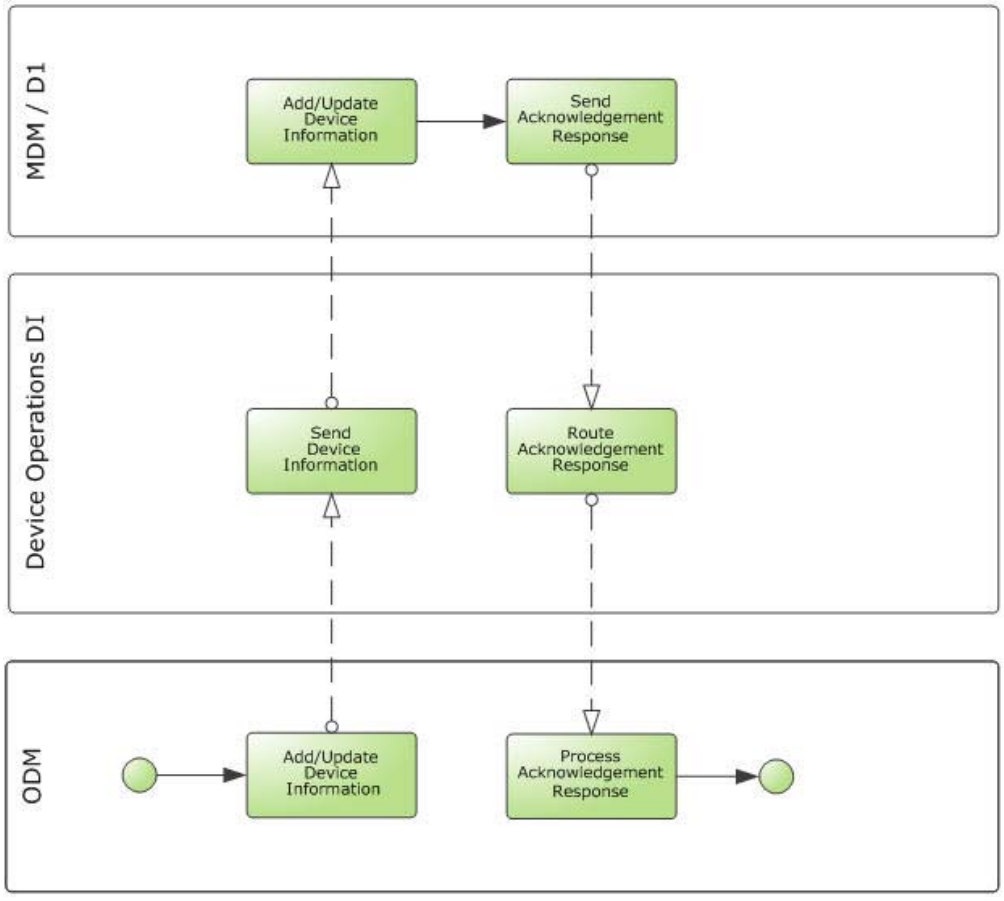
2.2.1.1 Asset-Device Synchronization

Oracle Utilities Operational Device Management puts Asset Device Sync message in the Asset Device Sync request queue.

Business Details

Asset-Device sync synchronizes device details required by Oracle Utilities Meter Data Management from Oracle Utilities Operational Device Management.

The following diagram provides a visual representation of this process:



Device Synchronization

Technical Details

Oracle Utilities Operational Device Management sends the device information as XML messages. These messages are transformed by the integration layer, and then sent to Oracle Utilities Meter Data Management. Oracle Utilities Meter Data Management sends an acknowledgement to the integration so that it can be transformed and sent to Oracle Utilities Operational Device Management.

Composites

- ODMMDMAssetDeviceSyncReqEBF - Asset-Device Sync request
- MDMODMAssetDeviceSyncRespEBF - Asset-Device Sync response

JMS Queues

Queue Name	Description
ODMAssetSyncRequest	Oracle Utilities Operational Device Management Asset sync request Used by the integration layer to read incoming Asset sync messages from Oracle Utilities Operational Device Management.
ODMAssetSyncRequestError	Oracle Utilities Operational Device Management Asset sync request error Error queue for Oracle Utilities Operational Device Management Asset Sync request
ODMAssetSyncResponse	Oracle Utilities Operational Device Management Asset sync response Used by the integration layer to add transformed Asset Sync response messages from Oracle Utilities Meter Data Management. Also any integration business errors will be sent to this queue.
ODMAssetSyncResponseError	Oracle Utilities Operational Device Management Asset sync response error Error queue for Oracle Utilities Operational Device Management Asset Sync response
MDMDeviceSyncRequest	Oracle Utilities Meter Data Management Device sync request Used by the integration to add transformed Device Sync request messages.
MDMDeviceSyncRequestError	Oracle Utilities Meter Data Management Device sync request error Error queue for Oracle Utilities Meter Data Management Device Sync request error
MDMDeviceSyncResponse	Oracle Utilities Meter Data Management Device sync response Used by the integration to read incoming Device Sync response messages from Oracle Utilities Meter Data Management.
MDMDeviceSyncResponseError	Oracle Utilities Meter Data Management Device Sync Response Error Error queue for Oracle Utilities Meter Data Management Device Sync response error

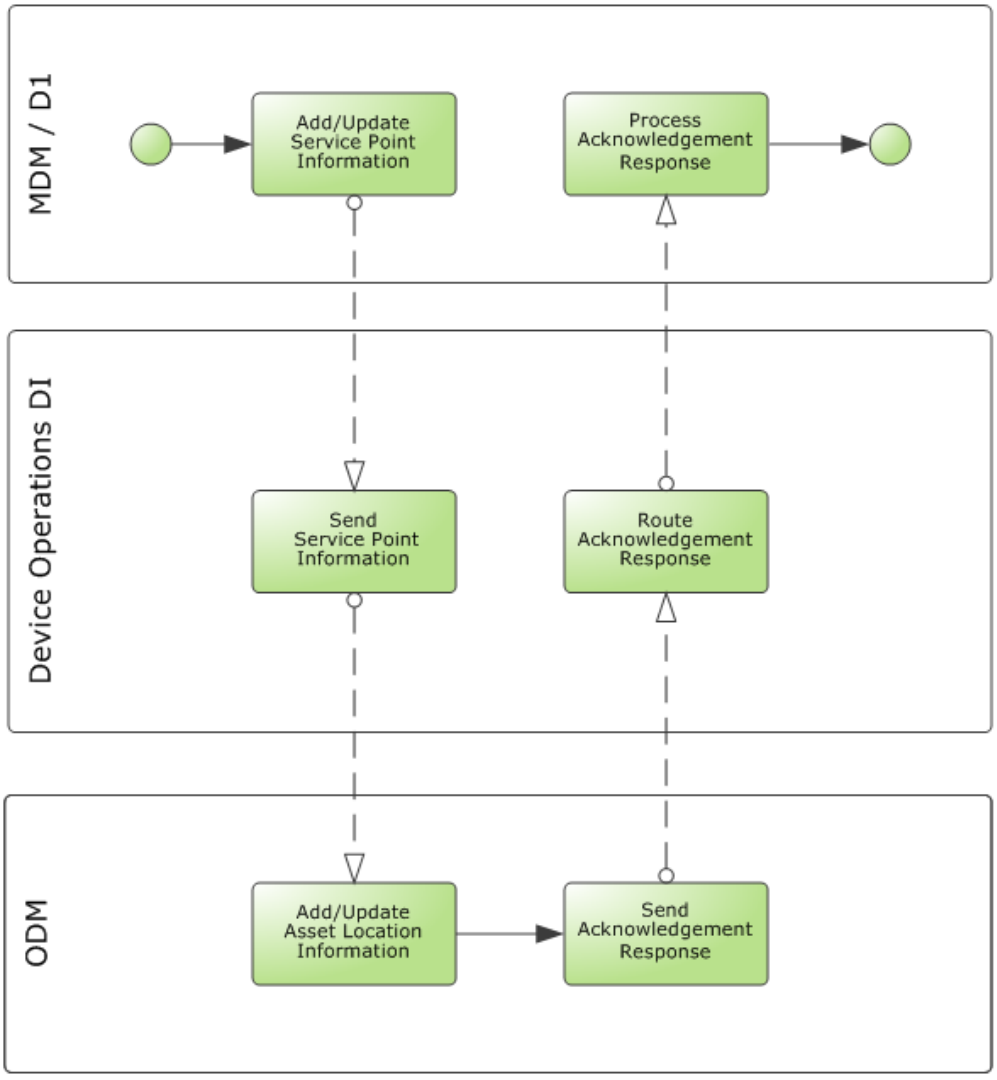
2.2.1.2 SP - Location Synchronization

Oracle Utilities Meter Data Management puts Service Point Sync message in the Service Point Sync request queue.

Business Details

This process synchronizes service point details required by Oracle Utilities Operational Device Management from Oracle Utilities Meter Data Management.

The following diagram provides a visual representation of this process:



Service Point Synchronization

Technical Details

Oracle Utilities Meter Data Management sends the service point information in the form of XML messages. These messages are transformed by the integration layer to node information, and then sent to Oracle Utilities Operational Device Management. Oracle Utilities Operational Device Management sends an acknowledgement to the integration, so that it can be transformed and sent to Oracle Utilities Meter Data Management.

> **Note**: Location information is stored as Node information in Oracle Utilities Operational Device Management

Composites

- MDMODMSPNodeSyncReqEBF - SP-Location sync request
- ODMMDMSPNodeSyncRespEBF - SP-Location sync response

JMS Queues

Queue Name	Description
MDMSPSyncRequest	Oracle Utilities Meter Data Management Service Point sync request Used by the integration layer to read incoming service point sync messages from Oracle Utilities Meter Data Management.
MDMSPSyncRequestError	Oracle Utilities Meter Data Management Service Point sync request error Error queue for Oracle Utilities Meter Data Management Service Point sync request
MDMSPSyncResponse	Oracle Utilities Meter Data Management Service Point sync response Used by the integration layer to add transformed service point sync response messages from Oracle Utilities Operational Device Management. Also any integration business errors will be sent to this queue.
MDMSPSyncResponseError	Oracle Utilities Meter Data Management Service Point sync response error Error queue for Oracle Utilities Meter Data Management Service Point sync response
ODMNodeSyncRequest	Oracle Utilities Operational Device Management Node sync request Used by the integration to add transformed Node sync request messages.
ODMNodeSyncRequestError	Oracle Utilities Operational Device Management Node sync request error Error queue for Oracle Utilities Operational Device Management Node sync request error

Queue Name	Description
ODMNodeSyncResponse	Oracle Utilities Operational Device Management Node sync response Used by the integration to read incoming Node sync response messages from Oracle Utilities Operational Device Management.
ODMNodeSyncResponseError	Oracle Utilities Operational Device Management Node sync response error Error queue for Oracle Utilities Operational Device Management Node sync response error

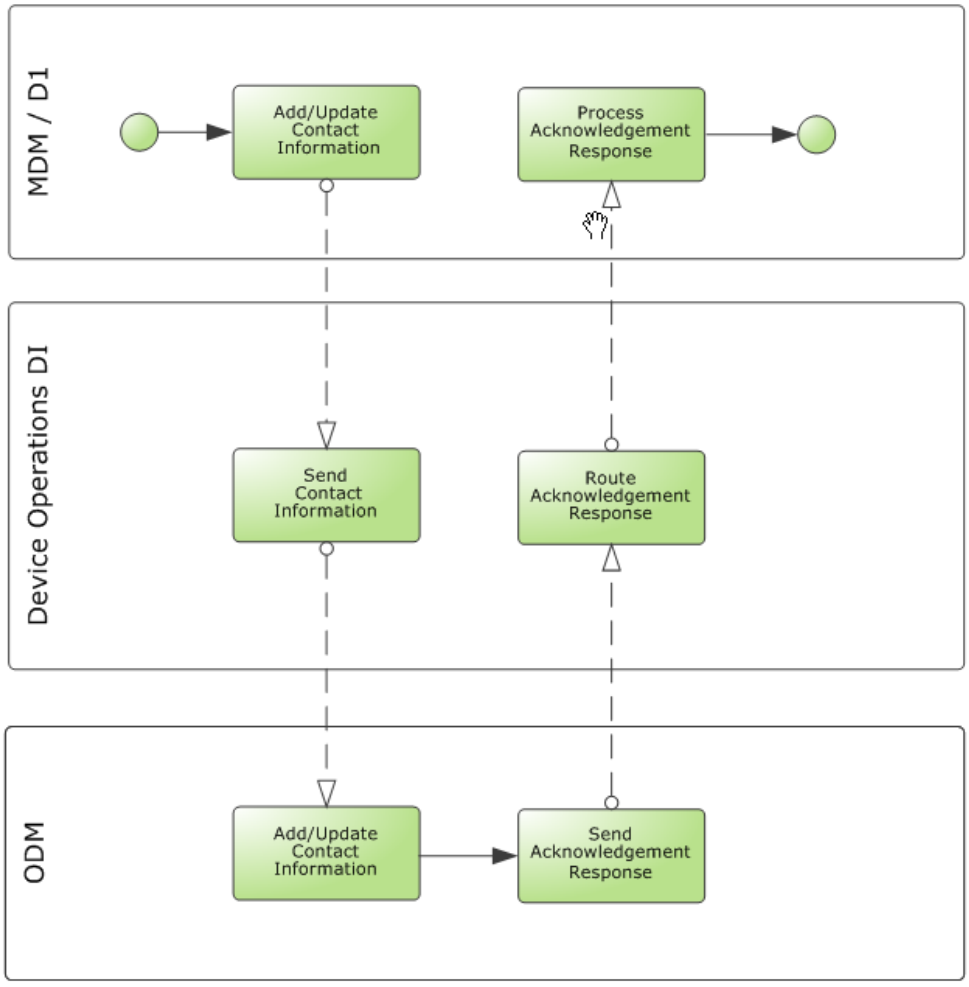
2.2.1.3 Contact Synchronization

Oracle Utilities Meter Data Management puts Contact sync message in the Contact Sync request queue.

Business Details

This process synchronizes contact details needed for Oracle Utilities Operational Device Management from Oracle Utilities Meter Data Management.

The following diagram provides a visual representation of this process:



Contact Synchronization

Technical Details

Oracle Utilities Meter Data Management sends the contact information in the form of XML messages. These messages are transformed by the integration layer, and then sent to Oracle Utilities Operational Device Management. Oracle Utilities Operational Device Management sends an acknowledgement to the integration, so that it can be transformed and sent to Oracle Utilities Meter Data Management.

Composites

- MDMODMContactSyncReqEBF - Contact sync request
- ODMMDMContactSyncRespEBF - Contact sync response

JMS Queues

Queue Name	Description
MDMContactSyncRequest	Oracle Utilities Meter Data Management Contact sync request Used by the integration layer to read incoming contact sync messages from Oracle Utilities Meter Data Management.
MDMContactSyncRequestError	Oracle Utilities Meter Data Management Contact sync request error Error queue for Oracle Utilities Meter Data Management Contact sync request
MDMContactSyncResponse	Oracle Utilities Meter Data Management Contact sync response Used by the integration layer to add transformed contact sync response messages from Oracle Utilities Operational Device Management. Also any integration business errors will be sent to this queue.
MDMContactSyncResponseError	Oracle Utilities Meter Data Management Contact sync response error Error queue for Oracle Utilities Meter Data Management Contact sync response
ODMContactSyncRequest	Oracle Utilities Operational Device Management Contact sync request Used by the integration to add transformed Contact sync request messages.
ODMContactSyncRequestError	Oracle Utilities Operational Device Management Contact sync request error Error queue for Oracle Utilities Operational Device Management Contact sync request error
ODMContactSyncResponse	Oracle Utilities Operational Device Management Contact sync response Used by the integration to read incoming Contact sync response messages from Oracle Utilities Operational Device Management.

Queue Name	Description
ODMContactSyncResponseError	Oracle Utilities Operational Device Management Contact sync response error Error queue for Oracle Utilities Operational Device Management Contact sync response error

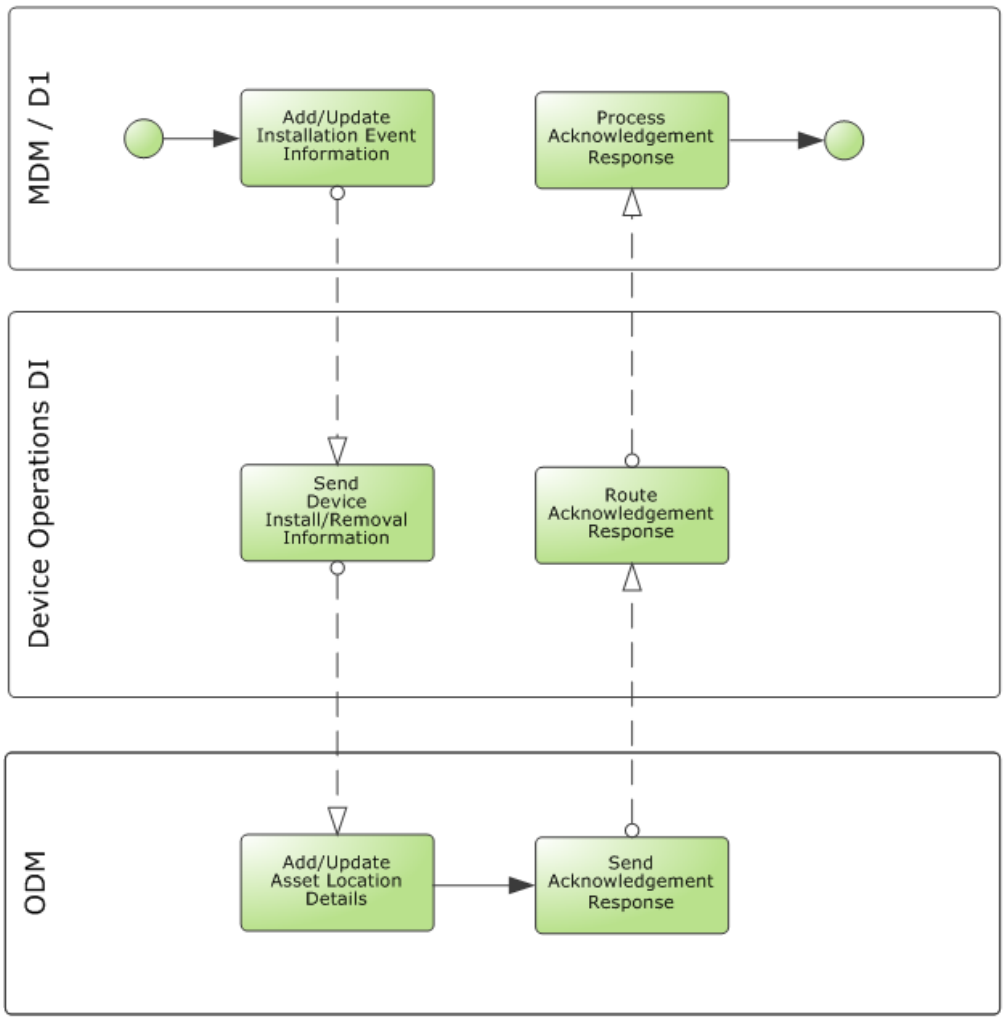
2.2.1.4 Install Event – Asset Location Synchronization

Oracle Utilities Meter Data Management inserts the Install Event Sync message in the Install Event Sync request queue then sends the Install Event Sync information in the form of an xml message which will be transformed by the integration to Asset Location message and sent to Oracle Utilities Operational Device Management. Any business errors are reported back to Oracle Utilities Meter Data Management. For Technical errors, the message can be resent from the integration layer to Oracle Utilities Operational Device Management.

Business Details

This process synchronizes asset location details required by Oracle Utilities Operational Device Management from install event in Oracle Utilities Meter Data Management.

The following diagram provides a visual representation of this process:



Device Location Synchronization

Technical Details

Oracle Utilities Meter Data Management sends the install event information in the form of XML messages. These messages are transformed by the integration layer to asset location format, and then sent to Oracle Utilities Operational Device Management. Oracle Utilities Operational Device Management sends an acknowledgement to the integration, so that it can be transformed and sent to Oracle Utilities Meter Data Management.

> **Note**: Asset Location information is stored as Asset Location information in Oracle Utilities Operational Device Management.

Composites

- MDMODMInstallEventAssetNodeSyncReqEBF - Install Event – Asset Location sync request
- ODMMDMInstallEventAssetNodeSyncRespEBF - Install Event – Asset Location sync response

JMS Queues

Queue Name	Description
MDMInstallEventSyncRequest	Oracle Utilities Meter Data Management Install Event sync request Used by the integration layer to read incoming install event sync messages from Oracle Utilities Meter Data Management.
MDMInstallEventSyncRequestError	Oracle Utilities Meter Data Management Install Event sync request error Error queue for Oracle Utilities Meter Data Management install event sync request
MDMInstallEventSyncResponse	Oracle Utilities Meter Data Management Install Event sync response Used by the integration layer to add transformed install event sync response messages from Oracle Utilities Operational Device Management. Also any integration business errors will be sent to this queue.
MDMInstallEventSyncResponseError	Oracle Utilities Meter Data Management Install Event sync response error Error queue for Oracle Utilities Meter Data Management install event sync response
ODMAssetNodeSyncRequest	Oracle Utilities Operational Device Management Asset location sync request Used by the integration to add transformed asset location sync request messages.
ODMAssetNodeSyncRequestError	Oracle Utilities Operational Device Management Asset location sync request error Error queue for Oracle Utilities Operational Device Management asset location sync request error

Queue Name	Description
ODMAssetNodeSyncResponse	Oracle Utilities Operational Device Management Asset Location sync response Used by the integration to read incoming asset location sync response messages from Oracle Utilities Operational Device Management.
ODMAssetNodeSyncResponseError	Oracle Utilities Operational Device Management Asset Node sync response error Error queue for Oracle Utilities Operational Device Management asset location sync response error

2.2.2 Prerequisites

All participating applications namely the Oracle Utilities Operational Device Management, Oracle Utilities Meter Data Management, and Oracle Service-Oriented Architecture Suite must be installed, set up, and working properly.

Part 2

Implementing the Integration Product

This section provides details on how to configure the participating applications and middleware layer for this AIA direct integration. It also includes information on error handling, monitoring, customization options, and data mapping.

This section contains the following chapters:

- Configuring the Integration
- Monitoring and Troubleshooting
- Customization Options

Chapter 3

Configuring the Integration

This section provides details regarding the configuration settings required for the integration, and also discusses in detail the following:

- Integration Configuration Checklist
- Data Synchronization
- Setting up Oracle Utilities Operational Device Management
- Setting Up Oracle Utilities Meter Data Management
- Setting up the Process Integration

3.1 Integration Configuration Checklist

Extensive configuration is required to implement the integration between Oracle Utilities Operational Device Management and Oracle Utilities Meter Data Management. This section provides a list of configuration tasks that may be used as a reference or roadmap.

- Oracle Utilities Operational Device Management Configuration
- Oracle Utilities Meter Data Management Configuration
- Integration Product Configuration

3.1.1 Oracle Utilities Operational Device Management Configuration

For integrating Oracle Utilities Operational Device Management, the following setup and configuration are required. These are explained in detail in the table below:

- Configure Admin Data Tables
- Configure System Data Tables
- JMS Configuration
- XAI Configuration

Oracle Utilities Operational Device Management Configuration Details

Task	Description
Configure Admin Data Tables	
Admin Data Setup	There are admin tables that are essential to the integration, for example: Asset Type, Specification, etc. These admin tables must be setup in Oracle Utilities Operational Device Management and any corresponding DVMs updated.
MDM Navigation Setup	Configure the MDM Integration master configuration. The Oracle Utilities Meter Data Management URL value must contain the valid Oracle Utilities Meter Data Management URL
Configure System Data Tables	
Sync Read Business Objects	Define the read BOs used to build the initial/final sync snapshot
Sync Data Area	Define the data area that holds the required elements
Sync Pre-processing Algorithm	Configure the pre-processing algorithms that create the initial snapshot for the various entities
Timeout Algorithm	Configure the timeout algorithm(s) that are used in sync, usage and corrected read processing
To Do Creation Algorithm	Configure the To Do entry algorithm(s) that are used to notify users of anomalies
Sync Request Business Object	Setup business objects that define the behavior of the outbound sync requests (link pre-processing, timeout and To Do algorithms, etc.)
Business Objects	Update the business objects that your implementation wants to create sync requests for (link audit algorithms and specify option values for sync request BO)

Task	Description
Extendable Lookups	Some extendable lookups were created to hold the values as defined in the external system being integrated with. Configure the values for these extendable lookups.
Lookup Field	Configure the required values for this lookup field.
Menu Setup	Enable the links from Oracle Utilities Operational Device Management to MDM
Batch Scheduling	Note the batch jobs involved in the sync and billing processes and ensure that these are incorporated in your scheduler accordingly
JMS Configuration	
JMS Setup	Configure the JMS to receive JMS messages from the integration layer
XAI Configuration	
XAI Setup	Configure JNDI server, JMS connection and queue, XAI sender, outbound message types and external systems for the integration

3.1.2 Oracle Utilities Meter Data Management Configuration

For integrating Oracle Utilities Meter Data Management, you need the following setup and configurations:

- Configure Admin Data Tables
- Configure System Data Tables
- JMS Configuration
- XAI Configuration

Oracle Utilities Meter Data Management Configuration Details

Task	Description
Configure Admin Data Tables	
Admin Data Setup	There are admin tables that are essential to the integration, for example: SP Type, US Type, Service Type, UOM, TOU, SQI, Device Type, etc. These admin tables must be setup in Oracle Utilities Meter Data Management and the corresponding DVMs updated.
MDM Navigation Setup	Configure the General System Configuration feature configuration. The Oracle Utilities Operational Device Management URL option value must contain the valid Oracle Utilities Operational Device Management URL.
Configure System Data Tables	
Sync Business Objects	These are the inbound sync business objects that are used to add or update the data in Oracle Utilities Meter Data Management. Only the ongoing sync business objects need further setup to define the Outbound Message Type to be used for any acknowledgements sent back to Oracle Utilities Operational Device Management.

Task	Description
BO Algorithms	These are Oracle Utilities Operational Device Management-specific algorithms that need to be plugged into the sync BOs. A BPA script is provided to for your convenience.
Extendable Lookups	Some extendable lookups were created to hold the values as defined in the external system being integrated with. Configure the values for these extendable lookups.
Lookup Field	Configure the required values for this lookup field.
MO Audit Algorithms	Configure the MO Audit algorithms responsible for sync request creation
Maintenance Objects	Update the maintenance objects that your implementation wants to create sync requests for (link audit algorithms and specify option values for sync request BO)
Menu Setup	Enable the links from Oracle Utilities Meter Data Management to Oracle Utilities Operational Device Management.
Batch Scheduling	Note the batch jobs involved in the sync and billing processes and ensure that these are incorporated in your scheduler accordingly.
JMS Configuration	
JMS Setup	Configure the JMS to receive JMS messages from the integration layer.
XAI Configuration	
XAI Setup	Configure JNDI server, JMS connection and queue, XAI sender, outbound message types and external systems for the integration.

3.1.3 Integration Product Configuration

Integration Product Configuration involves the following:

Integration Product Configuration Details

Task	Description
Setting Configuration Properties	Update ConfigurationProperties.xml file
Setting System Properties	Set Module Configurations properties that are shared by multiple integration flows and Service Configurations properties that are used by a specific BPEL process.
Domain Value Maps	Set Domain value maps (DVMs) to map codes and other static values across applications.
Error Handling	Setup error notification

3.2 Data Synchronization

For the purpose of this integration, Oracle Utilities Meter Data Management serves as the database of record for contacts, service points, and meter device connections while Oracle Utilities Operational Device Management manages assets (meters and components). Any addition of asset data in Oracle Utilities Operational Device Management is communicated to Oracle Utilities Meter Data Management where, based on a template ID, device, device configuration, and measuring component records are associated. Once these devices are installed at service points in MDM, the install information is communicated to Oracle Utilities Operational Device Management. Service point and contact information are necessary for the communication of install information and these are synchronized separately from Oracle Utilities Meter Data Management to Oracle Utilities Operational Device Management.

3.3 Setting up Oracle Utilities Operational Device Management

To set up Oracle Utilities Operational Device Management for the integration, configure the following:

- Configure Admin Data Tables
- Configure System Data Tables
- JMS Configuration
- XAI Configuration

For more information on configuring and working with Oracle Utilities Operational Device Management, refer to the Oracle Utilities Operational Device Management user documentation.

The following sections provide details about how to configure these items.

> **Note**: Some configurations described may be required for general functionality and do not necessarily relate directly to the integration; however these are called out as particularly significant configuration items. The inclusion of such items does not mean that other general items that are not mentioned do not need to be configured.

3.3.1 Configure Admin Data Tables

This section describes the unique setup issues specifically related to configuring your system for the integration.

For more information about configuring Oracle Utilities Operational Device Management, refer to the Oracle Utilities Operational Device Management User Guide.

This section contains the following:

- Country
- Contact Type
- Asset Location Type
- Time Zone
- Asset Type
- Specification
- Master Configuration

3.3.1.1 Country

Create a country code in Oracle Utilities Operational Device Management.

The **Main** page is used to customize the fields and field descriptions that are displayed where addresses are used in the system. This ensures that all addresses conform to the customary address format and conventions of the particular country you have defined. The codes defined here must exactly match values in the DVM indicated.

Country Code Details for Oracle Utilities Operational Device Management

Navigation	Guideline	Corresponding DVM
Admin Menu > Country	Create the Country codes	MDM_ODM_Country

3.3.1.2 Contact Type

Create the required Contact Types. The codes defined here must exactly match values in the DVM indicated.

Contact Type Code Details for Oracle Utilities Operational Device Management

Navigation	Guideline	Corresponding DVM
Admin Menu > Contact Type	Create Contact Types	MDM_ODM_ContactType

3.3.1.3 Asset Location Type

Create the required Asset Location Types. The codes defined here must exactly match values in the DVM indicated.

Asset Location Type Code Details for Oracle Utilities Operational Device Management

Navigation	Guideline	Corresponding DVM
Admin Menu > Asset Location Type	Create Asset Location types	MDM_ODM_SPNodeType

3.3.1.4 Time Zone

Create the required time zones. The codes defined here must exactly match values in the DVM indicated.

Time Zone code for Oracle Utilities Operational Device Management

Navigation	Guideline	Corresponding DVM
Admin Menu > TimeZone	Create Timezone Types	MDM_ODM_TimeZone

3.3.1.5 Asset Type

Create the required Asset Types. The codes defined here must exactly match values in DVM.

Specification codes for Oracle Utilities Operational Device Management

Navigation	Guideline	Corresponding DVM
Admin Menu > Asset Type	Create Asset Types	MDM_ODM_AssetDevice Type

3.3.1.6 Specification

Create the required Specification codes. The codes defined here will be used by the Specification extendable lookup in Oracle Utilities Meter Data Management.

Specification codes for Oracle Utilities Operational Device Management

Navigation	Guideline	Corresponding DVM
Admin Menu > Specification	Create Specification codes	N/A

3.3.1.7 Master Configuration

Two master configurations, Master Data Synchronization Configuration and Seeder Sync Request Master Configuration, need to be configured. A BPA script is provided to prepopulate these with the information necessary to support the base objects included in the sync. Run W2-MDMPreMCf to accomplish this.

A third master configuration, MDM Integration Master Configuration, needs to be completed manually. The Oracle Utilities Meter Data Management URL, timeout threshold hours for outbound requests and the MO/Outbound Message Type combinations used by outbound requests are all defined here. The To Do Type to use for asset-location sync scenarios that require manual intervention is also defined here as well as the time out number of hours used by the outbound sync.

3.3.2 Configure System Data Tables

To configure the System Data tables, you need the following:

- Business Objects
- BO Algorithms
- Extendable Lookups
- Menus
- Batch Scheduling

3.3.2.1 Business Objects

This section describes unique setup issues specifically related to configuring your system for integration.

Business Object	Description
W1-OngoingSyncRequestContact W1-OngoingSyncRequestLocation W1-OngoingSyncRequestAssetNode	These business objects define the behavior of inbound sync requests. The schema elements define information required to maintain the master data in Oracle Utilities Operational Device Management. As part of sync request processing, an acknowledgement message is sent to the external system (either positive or negative). The "Outbound Message Type" BO option contains a reference to the outbound message BO to use for this purpose. The base package includes BO W1-MDMAssetSyncReqOutMsg to be used on the outbound message type configuration. Refer to Outbound Configuration in this section for more information

Business Object	Description
	The error states in the ongoing sync requests contain automatic To Do creation and automatic retry. The parameters relevant to these processes (To Do Retry Frequency and To Do Maximum Retries) are captured as BO status options. If your implementation needs to introduce your own values, simply add a higher sequenced row for the option you intend to modify. The algorithms will automatically use these values. The Create To Do algorithm (W1-CRE-SRITD) has been delivered to use the base package supplied To Do Type W1-SYRQI for this process. If your implementation needs to use a different To Do Type, you will need to configure your own algorithm and supply the value in its parameters. The monitor process on the initial states of these BOs may be removed by the implementation if immediate processing of the sync requests as they are received is desired.
W1-MDMAssetSyncRequest	This business object defines the behavior of an outbound sync request. The schema elements define the information needed by Oracle Utilities Meter Data Management to maintain device master data. The business object and data area necessary to build the sync snapshot are defined as BO options on the sync request BO. The BO W1-MDMAsset and DA W1-MDMAssetSnapshot are included in the base for use with these options. If your implementation needs to use your own snapshot BO or DA, simply add a higher sequenced row for the relevant option. The Post Service Script for Extract option allows your implementation to perform additional manipulation of the data during the building of the snapshot. The Create To Do algorithm (W1-CRE-SROTD) has been delivered to use the base package supplied To Do Type W1-SYRQO for this process. If your implementation needs to use a different To Do Type, you will need to configure your own algorithm and supply the value in its parameters.
W2-SmartMeter, W2-ManualMeter, W2-CommunicationComponent	These base asset BOs need to have the audit plug-in configured (see below) in order to create outbound sync requests. The sync request BO to use for Oracle Utilities Meter Data Management is defined as a BO Option on these BOs. The base includes W1-MDMAssetSyncRequest for this purpose.

3.3.2.2 BO Algorithms

BO Algorithm Types and Description

Algorithm Type	Description
W1-GCHG-CDCP	.This BO audit algorithm creates a sync request (if there is not already a Pending one in existence) for the particular record being modified. It retrieves the Sync Request BO option from the record's BO.
W1-EXTSYSRST	This monitor algorithm sets a timeout limit on the receipt of a response from the external system. It retrieves this timeout limit from the MDM Integration Master Configuration. Use the latter to define the number of hours your implementation wishes to wait for a response from Oracle Utilities Meter Data Management before transitioning the sync request into the Error state.

3.3.2.3 Extendable Lookups

Configure the following extendable lookups in Oracle Utilities Operational Device Management:

- "W2-SmartMeterConfiguration - Configure the values to be used by Oracle Utilities Meter Data Management to identify a template device. This value is used in Oracle Utilities Meter Data Management in conjunction with the Specification to select a template device from which to create the device configuration and measuring components. Both configuration and specification values need to be set up in the Specification extendable lookup in Oracle Utilities Meter Data Management.
- "W2-ManualMeterConfiguration - Configure the values to be used by Oracle Utilities Meter Data Management to identify a template device. This value is used in Oracle Utilities Meter Data Management in conjunction with the Specification to select a template device from which to create the device configuration and measuring components. Both configuration and specification values need to be set up in the Specification extendable lookup in Oracle Utilities Meter Data Management.

3.3.2.4 Menus

Ensure that users have access to the application service W1-GOTOMDM. This application service is defined on the BPA scripts that take a user from Oracle Utilities Operational Device Management into Oracle Utilities Meter Data Management. The context menu items appear on Asset.

3.3.2.5 Batch Scheduling

This is the batch process to run the sync request. It is a generic batch process that is used for different sync processes. It has a couple of parameters that can be used to control which sync request BOs to process.

The following batch processes are used for Initial Sync requests:

> **Note**: All sync requests for all objects being synchronized are expected to be present in the **Pending** state. The general process flow of the batches for the initial sync request is described below.

Batch Scheduling Details

Batch Code	Description
F1-SYNRQ	This batch process transitions all the sync requests out of the **PENDING** state.
F1-SAKRQ	This batch process pre-allocates the production key to each record and transitions all the sync requests out of the Transformed/Schema Validated state into the Key Allocated state.
W1-SIKCN	This batch process resolves any foreign keys within the schema as well as executes the validation algorithms on the target BOs. This batch code is for processing Contact initial sync requests.
W1-SILCN	This batch process loads the records for Contact into the production tables.
W1-SIKSP	This batch process resolves any foreign keys within the schema as well as executes the validation algorithms on the target BOs. This batch code is for processing SP initial sync requests.
W1-SILSP	This batch process loads the records for SP into the production tables.
W1-SILDD	This batch process transitions all initial sync requests into the Additional Processing state from the Loaded state. An algorithm in the Additional Processing state sends an acknowledgement back to the external system along with the production ID of the synchronized object in Oracle Utilities Operational Device Management.
W1-SIIER	This is the batch process to transition sync request out of the **ERROR** state.

Note: If any sync requests exist in the Validation Error state, run W1-SIIER to retry the data transformation/schema validation process (only after the errors have been investigated and resolved). If any sync requests exist in the Resolution/BO Validation Error state, run its respective W1-SIK* batch job (refer to the above table for proper suffix to use for each master data record being synchronized).

Batch Processes for Ongoing Sync Requests

Batch Code	Description
W1-SIOPE	This is the batch process to transition ongoing sync request out of the **PENDING** state. It is a generic batch process that is used for different sync processes. It has a couple of parameters that can be used to control which sync request BOs to process.
W1-SIOER	This is the batch process to transition ongoing sync request out of the **ERROR** state.

Batch Processes for Outbound Sync Requests

Batch Code	Description
F1-SYNRQ	Sync Request Monitor Process. This is the batch process to run the sync request. It is a generic batch process that is used for different sync processes. It has a couple of parameters that can be used to control which sync request BOs to process.

Batch Code	Description
F1-SYNIL	Generic Sync Request Initial Load - This process creates an initial sync request BO for a particular MO. Algorithm parameters for filtering records are provided, so implementations can further restrict the creation of initial sync requests to certain records within the MO.

Note: To generate initial sync requests, submit the F1-SYNIL batch job. (Refer to the previous section for the parameters necessary to set it up.) To transition Sync Requests out of the Pending state, run the F1-SYNRQ batch job.

For more information about the sync request process, the business objects, maintenance objects, and other components used for this process, refer to the *Oracle Utilities Framework User Guide* section "Data Synchronization."

3.3.3 JMS Configuration

This section describes the JMS configuration to be done in the Oracle Utilities Operational Device Management WebLogic server and in the Oracle Utilities Operational Device Management deployment XML files. Configure the JMS to receive JMS messages from the integration layer.

- WebLogic Server JMS Configuration
- Configuration File Changes

3.3.3.1 WebLogic Server JMS Configuration

To configure JMS the Oracle Utilities Operational Device Management WebLogic server, log in to the console using the URL http://<server_name>:<port_number>/console.

For example: http://ODMserver:7001/console

JMS Module

Create a new JMS Module in the WebLogic Console

To Create a JMS module use for remote queue configuration:

1. Open the WebLogic console and create a new JMS module.
2. Enter a meaningful name for the JMS module. This JMS module is used to create configurations which consume messages from remote WebLogic queues. For example: MDMIntegrationModule.

Foreign Server

Create a new Foreign server under the JMS module in the WebLogic console.

To create the Foreign Server used for remote queue configuration:

1. Enter the WebLogic console and select the JMS module created for the integration.
2. Create a Foreign server under the JMS module. Enter the following for the Foreign server:
   - **Name** – Indicates the name for the Foreign server. For example: ODMMDMForeignServer
   - **JNDI Initial Context Factory** – weblogic.jndi.WLInitialContextFactory

- **JNDI Connection URL** – Add the URL for the Integration SOA server. For example: t3://soaserver.com:8002
- **JNDI Properties Credential** – Password for the SOA server user
- **JNDI Properties** - java.naming.security.principal=<SOA Server user> For example: weblogic

3. Under the Foreign server, create a foreign destination for each remote queue.
    - **Name** – Indicates the name of the foreign destination
    - **Local JNDI Name** - Add a local JNDI name for the Integration Queue. Local JNDI name is later added manually as part of configuration in weblogic-ejb-jar.xml --> <weblogic-enterprise-bean> --> <message-driven-descriptor> --> <destination-jndi-name>.
    - **Remote JNDI Name** – JNDI name of the queue on the Integration SOA server. Few examples are as follows: For each integration point, one destination is created.
    - **JNDI Properties Credential** – Password for the SOA server user
    - **JNDI Properties** - java.naming.security.principal=<SOA Server user> For example: weblogic

Asset Sync Foreign Destination Remote Queue Details

Destination Name	Local JNDI Name	Remote JNDI Name
ODMAssetSyncResponse	jms/LocalODMAssetSyncResponse	jms/ODM-MDM/ ODMAssetSyncRespon se

SP Location Sync Foreign Destination Remote Queue Details

Destination Name	Local JNDI Name	Remote JNDI Name
ODMNodeSyncRequest	jms/ LocalODMNodeSyncRequest	jms/ODM-MDM/ ODMNodeSyncRequest

Contact Sync Foreign Destination Remote Queue Details

Destination Name	Local JNDI Name	Remote JNDI Name
ODMContactSyncRequest	jms/ LocalODMContactSyncReq uest	jms/ODM-MDM/ ODMContactSyncRequest

Asset Location Sync Foreign Destination Remote Queue Details

Destination Name	Local JNDI Name	Remote JNDI Name
ODMAssetNodeSyncRe quest	jms/ LocalODMAssetNodeSyncRe quest	jms/ODM-MDM/ ODMAssetNodeSyncRequest

4. Under the Foreign server, create a Remote Connection Factory
    - **Name** – Indicates the name of the remote connection factory.
    - **Local JNDI Name** – Add a local JNDI name to the Integration Connection Factory. This JNDI name is added manually later as part of configuration in WebLogic-ejb-jar.xml --> <weblogic-enterprise-bean> --> <message-driven-descriptor> --> <connection-factory-jndi-name>.
    - **Remote JNDI Name** – JNDI name of the JMS Connection factory on the Integration SOA server. For example:

Remote Connection Factory Details

Connection Factory Name	Local JNDI Name	Remote JNDI Name
ODMMDMCF	jms/LocalODMMDMCF	jms/ODM-MDM/ ODMMDMCF

3.3.3.2 Configuration File Changes

To configure file changes you need the following:

- **Configure Message Driven Beans (MDB)**: It is recommended that you use the Oracle Utilities Operational Device Management template and CM (Customer Modification) feature to make changes to these configuration files. This ensures that your modifications cannot be overwritten by future application patches.
- **Modify files**: ejb-jar.xml and ejb-weblogic-jar.xml
- **Location**: Oracle Utilities Operational Device Management Enterprise Archive (EAR) file
    - The Oracle Utilities Operational Device Management configuration files, ejb-jar.xml and ejb-weblogic-jar.xml, must be modified to configure Message Driven Beans (MDB). MDBs which receive messages from the integration queues. These files are part of the Oracle Utilities Operational Device Management Enterprise Archive (EAR) file.
    - The Oracle Operational Device Management application needs to be redeployed after these changes are made.
- **Managing Configuration Files**: Configuration files such as config.xml, ejb-jar.xml and ejb-weblogic-jar.xml are managed through template configuration files which reside in the environment's templates directory. When the initialSetup.sh script is executed, environment specific information is combined with the template to create the target file which is then deployed to the correct location. When the environment is started up (spl.sh start) the changes are automatically deployed to WebLogic.
- **Extending existing templates**: It is possible to extend existing templates with the use of Include template file(s) in the same location as the existing template. Using #ouaf_user_exit within the target template that will be extended, additional configuration from the include template will be processed and appended to the target template where the #ouaf_user_exit is present.

- **Enabling changes for the integration**: To enable your changes for integration with Oracle Utilities Operational Device Management it is recommended that you first make a "CM" copy of the existing template and make your changes to the CM version. If there are any problems with starting the application it is a simple process to delete the CM versions of the files and rerun the initial setup to regenerate and redeploy the original versions.
  If you make CM versions of the template files and later install a patch which updates the base template, the CM version will not be updated.

  Note: Working examples of the configuration files are available for download from My Oracle Support in patch number 14668718 - ODM-MDM INTEGRATION CONFIGURATION EXAMPLES. Before installing the examples, read the Product Fix design document included in the patch for more information.

  Note: Working examples of the configuration files are available for download from My Oracle Support in patch number 14668718 - ODM-MDM INTEGRATION CONFIGURATION EXAMPLES. Before installing the examples, read the Product Fix design document included in the patch for more information.

To create MDB to receive messages from the Oracle Utilities Operational Device Management inbound queue:

1. Create a new MDB to receive messages from each integration inbound queue. For simplicity, we refer to the names of the target configuration files in the following examples.
   However, you should make your changes in the templates/cm_<target file>.include version of the file and then execute initalSetup.sh (Unix) or initalSetup.cmd (Windows) to deploy the generated file.
2. Create an MDB for each Oracle Utilities Operational Device Management inbound queue to receive messages and invoke the Oracle Utilities Operational Device Management service.
3. Create or modify cm_ejb-jar.xml.wls.jms_1.include, cm_ejb-jar.xml.wls.jms_2.include, cm_weblogic-ejb-jar.xml.jms.include, cm_config.xml.jms.include, or cm_config.xml.win.jms.include (for Windows systems only) files to configure the MDBs.
   - Add the <message-driven> and <container-transaction> tag for each inbound queue in the ejb-jar.xml.
   - cm_ejb-jar.xml.wls.jms_1.include example:

```
<!--Contact sync integration point -->
<message-driven>
<description>MDB for ODMContactSyncRequest</description>
<display-name>ODMContactSyncRequest</display-name>
<ejb-name>ODMContactSyncRequest</ejb-name>
<ejb-class>com.splwg.ejb.mdb.MessageProcessor</ejb-class>
<messaging-type>javax.jms.MessageListener</messaging-type>
<transaction-type>Bean</transaction-type>
<message-destination-type>javax.jms.Queue</message-destination-type>
</message-driven>
<!--SP - Location sync integration point -->
<message-driven>
<description>MDB for ODMNodeSyncRequest</description>
<display-name>ODMNodeSyncRequest</display-name>
<ejb-name>ODMNodeSyncRequest</ejb-name>
```

```
<ejb-class>com.splwg.ejb.mdb.MessageProcessor</ejb-class>
<messaging-type>javax.jms.MessageListener</messaging-type>
<transaction-type>Bean</transaction-type>
<message-destination-type>javax.jms.Queue</message-destination-type>
</message-driven>
<!--Install Event – Asset Location sync integration point -->
<message-driven>
<description>MDB for ODMAssetNodeSyncRequest</description>
<display-name>ODMAssetNodeSyncRequest</display-name>
<ejb-name>ODMAssetNodeSyncRequest</ejb-name>
<ejb-class>com.splwg.ejb.mdb.MessageProcessor</ejb-class>
<messaging-type>javax.jms.MessageListener</messaging-type>
<transaction-type>Bean</transaction-type>
<message-destination-type>javax.jms.Queue</message-destination-type>
</message-driven>
<!--Asset-Device sync integration point -->
<message-driven>
<description>MDB for ODMAssetSyncResponse</description>
<display-name>ODMAssetSyncResponse</display-name>
<ejb-name>ODMAssetSyncResponse</ejb-name>
<ejb-class>com.splwg.ejb.mdb.MessageProcessor</ejb-class>
<messaging-type>javax.jms.MessageListener</messaging-type>
<transaction-type>Bean</transaction-type>
<message-destination-type>javax.jms.Queue</message-destination-type>
</message-driven>
```

cm_ejb-jar.xml.wls.jms_2.include example:

```
<!--Contact sync integration point -->
<container-transaction>
<method>
<ejb-name>ODMContactSyncRequest</ejb-name>
<method-name>onMessage</method-name>
</method>
<trans-attribute>NotSupported</trans-attribute>
</container-transaction>
<!--SP - Location sync integration point -->
<container-transaction>
<method>
<ejb-name>ODMNodeSyncRequest</ejb-name>
<method-name>onMessage</method-name>
</method>
<trans-attribute>NotSupported</trans-attribute>
</container-transaction>
<!--Install Event – Asset Location sync integration point -->
<container-transaction>
<method>
<ejb-name>ODMAssetNodeSyncRequest</ejb-name>
<method-name>onMessage</method-name>
</method>
<trans-attribute>NotSupported</trans-attribute>
</container-transaction>
<!--Asset-Device sync integration point -->
<container-transaction>
<method>
<ejb-name>ODMAssetSyncResponse</ejb-name>
<method-name>onMessage</method-name>
</method>
<trans-attribute>NotSupported</trans-attribute>
</container-transaction>
```

- Modify the cm_weblogic-ejb-jar.xml.jms.include file. Add the <weblogic-enterprise-bean> tag for each inbound queue.
- Add a security role with role cisusers.

The references in <weblogic-enterprise-bean> tag.are as follows:

- <ejb-name> - MDB name given in ejb-jar.xml.
- <destination-jndi-name> - JNDI name provided in JMS module -> Foreign server -> Foreign destination -> Local JNDI name.
- <connection-factory-jndi-name> - JNDI name provided in JMS module ' Foreign server ' Remote Connection Factory ' Local JNDI name.
- cm_weblogic-ejb-jar.xml.jms.include example:

```
<!--Contact sync integration point -->
<weblogic-enterprise-bean>
<ejb-name>ODMContactSyncRequest</ejb-name>
<message-driven-descriptor>
<pool>
<max-beans-in-free-pool>5</max-beans-in-free-pool>
<initial-beans-in-free-pool>1</initial-beans-in-free-pool>
</pool>
<destination-jndi-name>jms/LocalODMContactSyncRequest</destination-
jndi-name>
<connection-factory-jndi-name>jms/LocalODMMDMCF</connection-factory-
jndi-name>
</message-driven-descriptor>
</weblogic-enterprise-bean>

<!--SP - Location sync integration point -->
<weblogic-enterprise-bean>
<ejb-name>ODMNodeSyncRequest</ejb-name>
<message-driven-descriptor>
<pool>
<max-beans-in-free-pool>5</max-beans-in-free-pool>
<initial-beans-in-free-pool>1</initial-beans-in-free-pool>
</pool>
<destination-jndi-name>jms/LocalODMNodeSyncRequest</destination-jndi-
name>
<connection-factory-jndi-name>jms/LocalODMMDMCF</connection-factory-
jndi-name>
</message-driven-descriptor>
</weblogic-enterprise-bean>

<!--Install Event – Asset Location sync integration point -->
<weblogic-enterprise-bean>
<ejb-name>ODMAssetNodeSyncRequest</ejb-name>
<message-driven-descriptor>
<pool>
<max-beans-in-free-pool>5</max-beans-in-free-pool>
<initial-beans-in-free-pool>1</initial-beans-in-free-pool>
</pool>
<destination-jndi-name>jms/LocalODMAssetNodeSyncRequest</destination-
jndi-name>
<connection-factory-jndi-name>jms/LocalODMMDMCF</connection-factory-
jndi-name>
</message-driven-descriptor>
</weblogic-enterprise-bean>
```

```
<!--Asset - Device sync integration point -->
<weblogic-enterprise-bean>
<ejb-name>ODMAssetSyncResponse</ejb-name>
<message-driven-descriptor>
<pool>
<max-beans-in-free-pool>5</max-beans-in-free-pool>
<initial-beans-in-free-pool>1</initial-beans-in-free-pool>
</pool>
<destination-jndi-name>jms/LocalODMAssetSyncResponse</destination-
jndi-name>
<connection-factory-jndi-name>jms/LocalODMMDMCF</connection-factory-
jndi-name>
</message-driven-descriptor>
</weblogic-enterprise-bean>
```

- cm_config.xml.jms.include/ cm_config.xml.win.jms.include example:

```
<jms-system-resource>
<name>ODMMDMIntegrationExampleModule</name>
<target>myserver</target>
<sub-deployment>
<name>ODMMDMIntegrationExample</name>
<target>myserver</target>
</sub-deployment>
<descriptor-file-name>jms/Module-for-MDM-integration-example-1-
jms.xml</descriptor-file-name>
</jms-system-resource>
```

3.3.4 XAI Configuration

The XML Application Integration (XAI) configuration includes the following:

- XAI JNDI Server
- XAI JMS Queue
- XAI JMS Connection
- XAI Sender
- Outbound Message Type
- External System

3.3.4.1 XAI JNDI Server

Create a new XAI JNDI server configured to communicate with the integration layer. It should point to the Integration SOA server.

To create the XAI JNDI server:

1. In the Admin menu, navigate to XAI JNDI server.
2. Enter the XAI JNDI server name. For example: **MDM_JNDI**
3. Enter the XAI JNDI server description For example: **ODM-MDM Integration server**
4. Enter the Provider URL in the format **t3//<SOA Server>: <SOA Port>**. For example: **t3://soaserver.us.oracle.com:8002**

3.3.4.2 XAI JMS Queue

Create a new XAI JMS queue for each integration queue where Oracle Utilities Operational Device Management sends messages.

To create XAI JMS queue:

1. In the Admin menu, navigate to XAI JMS queue.
2. Enter the following:
   - **XAI JMS Queue** – Refers to the queue name in Oracle Utilities Operational Device Management.
   - **Description** – Indicates queue description
   - **Queue Name** – Indicates the JNDI name of the queue on the Integration server. For example: jms/ODM-MDM/ODMContactSyncResponse
   - **Target Client Flag** – JMS
   - **XAI JNDI Server** – Select the XAI JNDI server created for integration.

     Note: You should only define the queues that Oracle Utilities Operational Device Management will be publishing or writing messages to.

Examples of the JMS queue that need to be setup for the different integration points are as follows:

Asset-Device Sync Integration Point Details

XAI JMS Queue	Description	Queue Name	Target Client Flag	XAI JNDI Server
AssSyncReq	Asset Sync Request	jms/ODM-MDM/ ODMAssetSync Request	JMS	MDM_JNDI

SP-Location Syn Integration Point Details

XAI JMS Queue	Description	Queue Name	Target Client Flag	XAI JNDI Server
NdeSyncRes	Node Sync Response	jms/ODM-MDM/ ODMNodeSync Response	JMS	MDM_JNDI

Contact Sync Integration Point Details

XAI JMS Queue	Description	Queue Name	Target Client Flag	XAI JNDI Server
ConSyncRes	Contact Sync Response	jms/ODM-MDM/ ODMContactSyncResponse	JMS	MDM_JNDI

Install Event - Asset Location Sync Integration Point Details

XAI JMS Queue	Description	Queue Name	Target Client Flag	XAI JNDI Server
ANSyncRes	Asset Node Sync Response	jms/ODM-MDM/ ODMAssetNodeSyncResponse	JMS	MDM_JNDI

3.3.4.3 XAI JMS Connection

Create a new XAI JMS connection used to connect to the integration queues.

To create XAI JMS Connection:

1. In the Admin menu, navigate to XAI JMS connection.
2. Enter the following:
   - **XAI JMS Connection** – Indicates the Connection name in Oracle Utilities Operational Device Management.
   - **Description** – Refers to the Connection description
   - **XAI JNDI Server** – Select the XAI JNDI server created for this integration (as described in the XAI JNDI Server section).
   - **JNDI ConnectionFactory** – JNDI name of the connection factory on the Integration server. For example: jms/ODM-MDM/ODMMDMCF

XAI JMS Connection Details

XAI JMS Connection	Description	XAI JNDI Server	JNDI Connection Factory
MDM2_CF	ODM MDM Integration Connection	MDM_JNDI	jms/ODM-MDM/ ODMMDMCF

3.3.4.4 XAI Sender

Create a new realtime XAI Sender for each Oracle Utilities Operational Device Management Outbound integration queue.

To create a XAI sender:

1. In the Admin menu, navigate to XAI Sender.
2. Enter a unique XAI sender and its description.
3. Populate the following values:
   - **XAI Sender** – Sender name in Oracle Utilities Operational Device Management.
   - **Description** – Sender description
   - **Invocation Type** – Real-time
   - **XAI Class** – RTJMSQSNDR (Realtime JMS Queue Sender)
   - **Active** - Select the checkbox.
   - **MSG Encoding** – UTF-8 message encoding

- **XAI JMS Connection** – XAI JMS connection created for the integration.
- **XAI JMS Queue** – XAI JMS queue created for the Oracle Utilities Operational Device Management outbound queue.

4. Select the Context tab and set values for the following context types:
   - JMS Message Type (Bytes(**Y**)/Text(**N**)) – **N**
   - **JMS User Name** – Indicates the user for the SOA server to be accessed.
   - J**MS User Password** – Refers to the Password for the SOA server to be accessed.

   Examples of the XAI sender that need to be setup for the different integration points are as follows:

XAI Sender Details for Asset-Device Integration Point

XAI Sender	Description	XAI JMS Connection	XAI JMS Queue
AssSyncReq	Asset Sync Request Sender	MDM2_CF	AssSyncReq

XAI Sender Details for SP-Location Sync Integration Point

XAI Sender	Description	XAI JMS Connection	XAI JMS Queue
NdeSyncRes	Node Sync Response	MDM2_CF	NdeSyncRes

XAI Sender Details for Contact Sync Integration Point

XAI Sender	Description	XAI JMS Connection	XAI JMS Queue
ConSyncRes	Contact Sync Response	MDM2_CF	ConSyncRes

Install Event - Asset Location Sync Integration

XAI Sender	Description	XAI JMS Connection	XAI JMS Queue
ANSyncRes	Asset Location Sync Response	MDM2_CF	ANSyncRes

3.3.4.5 Outbound Message Type

Create a new outbound message type for each Oracle Utilities Operational Device Management outbound integration queue.

To create new outbound message types:

1. In the Admin menu, navigate to Outbound Message Type.
2. Enter an outbound message type, description, and detailed description.
3. Select the outbound message business object created for a specific outbound queue.

   Examples of the Outbound Message Type that need to be setup for the different integration points are as follows:

Device - Asset Sync Integration Point Outbound Message

Outbound Message Type	Description	Business Object
WM-MDMASTSYN	Asset Sync Request - For Outbound Message	W1-MDMAssetSyncReqOutMsg

SP-Location Sync Integration Point Outbound Message

Outbound Message Type	Description	Business Object
WM-SP-SYNRSP	SP Sync Response - For Acknowledgment Message	W1-OutboundAcknowledgeMessage

Contact Sync Integration Point Outbound Message

Outbound Message Type	Description	Business Object
WM-CN-SYNRSP	Contact Sync Response - For Acknowledgment Message	W1-OutboundAcknowledgeMessage

Install Event - Asset Location Sync Integration Point Outbound Message

Outbound Message Type	Description	Business Object
WM-AN-SYNRSP	Asset-Location Sync Response - For Acknowledgment Message	W1-OutboundAcknowledgeMessage

3.3.4.6 External System

Create a new external system to support the integration.

To create the external system:

1. In the Admin menu, navigate to External System.
2. Enter a unique external system and description. For example: Name = MDM, Description = Oracle Utilities Meter Data Management
3. Set the Our Name in Their System field to ODM.
4. Associate the outbound message types created to the external system. For each outbound message type, set the following:
   - **Outbound Message Type** – Set the outbound message type created for Oracle Utilities Operational Device Management outbound queue.
   - **Processing Method** – Real-time
   - **XAI Sender** – Set the XAI sender created for the queue.
   - Message XSL
     For example: External System – MDM

External System Details

Outbound Message Type	Processing Method	XAI Sender	Message XSL
WM-MDMASTSYN	Real-time	AssSyncReq	W2-MDMODMRequestAddNamespace.xsl
WM-SP-SYNRSP	Real-time	NdeSyncRes	W2-MDMODMResponseAddNamespace.xsl
WM-CN-SYNRSP	Real-time	ConSyncRes	W2-MDMODMResponseAddNamespace.xsl
WM-AN-SYNRSP	Real-time	ANSyncRes	W2-MDMODMResponseAddNamespace.xsl

For more information regarding configuration guidelines, refer to the Oracle Utilities Operational Device Management documentation.

3.4 Setting Up Oracle Utilities Meter Data Management

Two Oracle Business Process Analysis (BPA) scripts are provided to aid in the setup of Oracle Utilities Meter Data Management in sync processing. Consequently, the amount of setup work to be done by an implementation is greatly reduced.

These scripts are intended to be executed only once prior to any customizations being made to any of the sync objects as noted below:

- Insert Oracle Utilities Operational Device Management-specific algorithms to Sync Request BOs
- D1-AddODMAlg - This script inserts transformation algorithms specific to the Oracle Utilities Operational Device Management integration into the sync request BOs. If your implementation needs to introduce additional transformation algorithms, they need to be inserted after this script is run.
- D1-ODMPreMcg- This script sets up data in the following master configurations:
  - Master Data Synchronization Configuration – contains the foreign key reference information used by framework to validate and/or resolve foreign keys in the master data sync requests.
  - Seeder Sync Request Master Configuration – contains information needed by the sync request seeder BO to determine the actual BO to instantiate. The information is keyed to external system, MO, and initial load indicator in the sync request.

The rest of the setup tasks in Oracle Utilities Meter Data Management include:

- Configure Admin Data Tables
- Configure System Data Tables
- JMS Configuration
- XAI Configuration

The following sections provide a general overview of these steps. For details, refer to the *Oracle Utilities Meter Data Management Installation and Configuration Guide*.

For more information on configuring and working with Oracle Utilities Meter Data Management, refer to *Oracle Utilities Meter Data Management Installation and Configuration Guide*.

The following sections provide details regarding the most significant configuration items:

3.4.1 Configure Admin Data Tables

This section describes the unique setup issues specifically related to configuring your system for the integration.

To configure Admin Data Tables, you need to define the following codes and types:

- Country
- Service Point Type
- Contact Type
- Device Type
- Master Configuration

For more information regarding configuring Oracle Utilities Meter Data Management, refer to the Oracle Utilities Meter Data Management User Guide.

3.4.1.1 Country

Create a country code in Oracle Utilities Meter Data Management.

The **Main** page is used to customize the fields and field descriptions that are displayed where addresses are used in the system. This ensures that all addresses conform to the customary address format and conventions of the particular country you have defined.

Country code for Oracle Utilities Meter Data Management

Navigation	Guideline	Corresponding DVM
Admin Menu > Country	Create the Country codes	MDM_ODM_Country

3.4.1.2 Service Point Type

Create the required SP Types. The codes defined here must exactly match values in the DVM indicated.

Service Point Type Code for Oracle Utilities Meter Data Management

Navigation	Guideline	Corresponding DVM
Admin Menu > Service Point Type	Create SP Types	MDM_ODM_SPNodeType

3.4.1.3 Contact Type

Create the required Contact Types. The codes defined here must exactly match values in the DVM indicated.

Contact Type Code for Oracle Utilities Meter Data Management

Navigation	Guideline	Corresponding DVM
Admin Menu > Contact Type	Create Contact Types	MDM_ODM_ContactType

3.4.1.4 Device Type

Create the required Device Types. The codes defined here must exactly match values in the DVM indicated.

Device Type Code for Oracle Utilities Meter Data Management

Navigation	Guideline	Corresponding DVM
Admin Menu > Device Type	Create Device Types	MDM_ODM_AssetDevice Type

3.4.1.5 Master Configuration

Two master configurations (Master Data Synchronization Configuration and Seeder Sync Request Master Configuration) need to be completed. A BPA script is provided to prepopulate these with the information necessary to support the base objects included in the sync. Run D1-ODMPreMcg to accomplish this.

A third master configuration (ODM Integration Master Configuration) needs to be completed manually. The ODM URL, timeout threshold hours for outbound requests and the MO/Outbound Message Type combinations used by outbound requests are all defined here.

3.4.2 Configure System Data Tables

To configure System Data Tables, you need the following:

- Business Objects
- BO Algorithms
- MO Algorithms
- Maintenance Object
- Extendable Lookups
- Lookups
- Menus
- Batch Scheduling

3.4.2.1 Business Objects

This section describes unique setup issues specifically related to configuring your system for integration.

Business Objects Details

Business Object	Description
D1-OngoingSyncRequestDevice	This business object defines the behavior of an ongoing sync request. The schema elements define information required to maintain the device master data in Oracle Utilities Meter Data Management. As part of sync request processing, an acknowledgement message is sent to the external system (either positive or negative). The "Outbound Message Type" BO option contains a reference to the outbound message BO to use for this purpose. The base package includes BO D1-OngoingSyncReqAckMsg to be used on the outbound message type configuration. Refer to Outbound Configuration in this section for more information The error states in the ongoing sync requests contain automatic To Do creation and automatic retry. The parameters relevant to these processes (To Do Retry Frequency and To Do Maximum Retries) are captured as BO status options. If your implementation needs to introduce your own values, simply add a higher sequenced row for the option you intend to modify. The algorithms will automatically use these values. The Create To Do algorithm (D1-TDCREATE) has been delivered to use the base package supplied To Do Type D1-SYNIN for this process. If your implementation needs to use a different To Do Type, you will need to configure your own algorithm and supply the value in its parameters. The monitor process on the initial states of these BOs may be removed by the implementation if immediate processing of the sync requests as they are received is desired.

Business Object	Description
D1-ODMContactSyncRequest D1-ODMSPSyncRequest D1-InstallEventSyncRequest	These business objects define the behavior of an outbound sync request. The schema elements define the information needed by Oracle Utilities Operational Device Management to maintain contact, SP, and asset-node master data. The business object and data area necessary to build the sync snapshot are defined as BO options on the sync request BO. The BOs D1-ContactODMBORead, D1-InstallEventBORead, and D1-ServicePointODMBORead as well as DAs D1-ODMContactBasedSnapshot, D1-ODMSPSnapshot, and D1-InstallEventSnapshot are included in the base for use with these options. If your implementation needs to use your own snapshot BO or DA, simply add a higher sequenced row for the relevant option. The Post Service Script for Extract option allows your implementation to perform additional manipulation of the data during the building of the snapshot. The Create To Do algorithm (D1-SRCRETODO) has been delivered to use the base package supplied To Do Type F1-SYNRQ for this process. If your implementation needs to use a different To Do Type, you will need to configure your own algorithm and supply the value in its parameters.

For more information about the sync request process, the business objects, maintenance objects, and other components used for this process, see the "Data Synchronization" section in *Oracle Utilities Framework User Guide*.

3.4.2.2 BO Algorithms

Oracle Utilities Operational Device Management-specific algorithms need to be plugged in on the sync request business objects. A BPA script is provided to plug-in these algorithms on to the sync BOs. This BPA script should be executed prior to any further customizations done on the sync BOs. It should only be executed once. Run D1-AddODMAlg (Insert ODM-specific algorithms to Sync Request BOs (MDF) to accomplish this.

3.4.2.3 MO Algorithms

Configure the MO Audit algorithms. MO Audit algorithms contain the logic to instantiate a sync request (as long as one does not already exist in the initial state for the MO-Primary Keys combination). A generic algorithm F1-GCHG-CDCP comes with the

base product and is plugged in on MOs that need to instantiate sync requests for the same MO. This algorithm instantiates the BOs defined in the Sync Request BO MO Option (see Maintenance Objects below). For MOs that have idiosyncratic logic, unique algorithms that contain this logic are used. (For example, changes to the Contact MO are only communicated to Oracle Utilities Operational Device Management if the Contact is linked to an SP.)

Algorithm Type	Description
D1-CONCDCSP	This algorithm instantiates a Contact-based sync request whenever a change to the Contact MO is detected and the Contact is associated with an SP. Define the sync request BO to be instantiated in the algorithm's parameters.

3.4.2.4 Maintenance Object

Configure the Maintenance Object algorithm type.

Maintenance Object Alogrithm Type

Algorithm Type	Description
D1-CONTACT	Specify the MO Audit algorithm configured in the previous section.
D1-SP	Specify the generic MO Audit algorithm D1-ODM-GCDCP. Also specify the D1-ODMSPSyncRequest BO in the Sync Request BO MO Option.
D1-INSTLEVT	Specify the generic MO Audit algorithm D1-ODM-GCDCP. Also specify the D1-InstallEventSyncRequest BO in the Sync Request BO MO Option.

3.4.2.5 Extendable Lookups

Configure the following extendable lookups in Oracle Utilities Meter Data Management:

- D1-OkToEnterLookup – Configure the corresponding values from Oracle Utilities Operational Device Management (values are yes or no). If values already exist (say, for a CIS integration), ensure that the values are mapped correctly using the DVM MDM_ODM_OkToEnterCode.

3.4.2.6 Lookups

Configure the following lookups in Oracle Utilities Meter Data Management:

- DL_LS_SL_FLG – Configure the corresponding values from Oracle Utilities Operational Device Management (values are Life Support or NA). If values already exist (say, for a CIS integration), ensure that the values are mapped correctly using the DVM MDM_ODM_LifeSupportSensitiveLoad.

3.4.2.7 Menus

Ensure that users have access to the application service D1-GOTOODM. This application service is defined on the BPA scripts that take a user from Oracle Utilities

Meter Data Management into Oracle Utilities Operational Device Management. The context menu items appear on Contact and SP.

3.4.2.8 Batch Scheduling

The delivered batch codes need no further setup in Oracle Utilities Meter Data Management. The following batch processes can be run from the Batch Submission page.

> **Note**: All sync requests for all objects being synchronized are expected to be present in the Pending state.

The general process flow of the batches for the initial sync request is described below:

Batch Processes for Initial Sync Requests

Batch Code	Description
F1-SYSRQ	This batch process transitions all the sync requests out of the **PENDING** state.
F1-SAKRQ	This batch process pre-allocates the production key to each record and transitions all the sync requests out of the Transformed/ Schema Validated state into the Key Allocated state.
D1-SIKDV	This batch process resolves any foreign keys within the schema as well as executes the validation algorithms on the target BOs. This batch code is for processing Device initial sync requests.
D1-SILDV	This batch process loads the records for Device into the production tables.
D1-SIIER	This is the batch process to transition sync request out of the **ERROR** state.

> **Note**: If any sync requests exist in the Validation Error state, run D1-SIIER to retry the data transformation/schema validation process (after the errors have been investigated and resolved). If any sync requests exist in the Resolution/BO Validation Error state, run its respective D1-SIK* batch job (refer to the above table for proper suffix to use for each master data record being synchronized).

Batch Code for Ongoing Sync Requests

Batch Code	Description
D1-SIOPE	This is the batch process to transition ongoing sync request out of the **PENDING** state. It is a generic batch process that is used for different sync processes. It has a couple of parameters that can be used to control which sync request BOs to process.
D1-SIOER	This is the batch process to transition ongoing sync request out of the **ERROR** state

Note: Depending on how sync requests are sent from Oracle Utilities Operational Device Management, it is possible that interdependent sync requests might be received out of order. If you wish to control the order of processing the ongoing sync requests within, you can either introduce your own batch controls to replace D1-SIOPE (each batch control will have the specific ongoing sync BO defaulted in the input parameter); or you can submit D1-SIOPE several times, each time specifying a different ongoing sync BO in the input parameter. Otherwise, you can let the built-in retry processing within the ongoing sync request life cycle resolve the error by running D1-SIOER.

Batch Processes for Usage Transaction Processing

Batch Code	Description
F1-SYNRQ	Sync Request Monitor Process. This is the batch process to run the sync request. It is a generic batch process that is used for different sync processes. It has a couple of parameters that can be used to control which sync request BOs to process.
F1-SYNIL	Generic Sync Request Initial Load - This process creates an initial sync request BO for a particular MO. Algorithm parameters for filtering records are provided, so implementations can further restrict the creation of initial sync requests to certain records within the MO.

3.4.3 JMS Configuration

This section describes the JMS configuration to be done in the Oracle Utilities Meter Data Management WebLogic server and in the Oracle Utilities Meter Data Management deployment XML files. The configuration described in this section is used for receiving JMS messages from the integration layer as shown in these topics:

- WebLogic Server JMS Configuration
- Configuration File Changes

3.4.3.1 WebLogic Server JMS Configuration

To configure JMS in the Oracle Utilities Meter Data Management WebLogic server log in to the console using the URL http://<server_name>:<port_number>/console.

For example: http://mdmserver:7001/console

JMS Module

Create a new JMS module in the WebLogic console **used for remote queue configuration.**.

To create a JMS module :

1. Open the WebLogic console and create a new JMS module.
2. Enter a meaningful name for the JMS module. This JMS module is used to create configurations which consume messages from remote WebLogic queues. For example: ODMIntegrationModule.

Foreign Server

Create a new Foreign server under the JMS module in the WebLogic console **used for remote queue configuration**.

To create the Foreign server:

1. Open the WebLogic console and select the JMS module created for integration.
2. Create a Foreign server under the JMS module.
3. Add the following:
   - **Name** – Indicates the name of the Foreign server. For example: MDMODMForeignServer
   - **JNDI Initial Context Factory – weblogic.jndi.WLInitialContextFactory**
   - **JNDI Connection URL** – Add the URL of Integration SOA server. For example: **t3://soaserver.com:8002**
   - **JNDI Properties Credential** – Password for the SOA server user.
   - **JNDI Properties** - **java.naming**.security.**principal=<SOA Server user>**. For example: weblogic
4. Under the Foreign server, create a Foreign destination for each remote queue:
   - Name – Name of foreign destination.
   - Local JNDI Name – Add a local JNDI name for the Integration queue. Local JNDI name is later added manually as part of configuration in the WebLogic-ejb-jar.xml -> <weblogic-enterprise-bean> -> <message-driven-descriptor> -> <destination-jndi-name>.
   - Remote JNDI Name – JNDI name of the queue on the Integration SOA server. Few examples are as follows. For each Integration point, one destination is created

Device -Asset Sync Foreign Destination Details

Destination Name	Local JNDI Name	Remote JNDI Name
MDMDeviceSyncRequest	jms/LocalMDMDeviceSyncRequest	jms/ODM-MDM/MDMDeviceSyncRequest

SP-Location Sync Foreign Destination Details

Destination Name	Local JNDI Name	Remote JNDI Name
MDMSPSyncResponse	jms/LocalMDMSPSyncResponse	jms/ODM-MDM/MDMSPSyncResponse

Contact Sync Foreign Destination Details

Destination Name	Local JNDI Name	Remote JNDI Name
MDMContactSyncResponse	jms/LocalMDMContactSyncResponse	jms/ODM-MDM/MDMContactSyncResponse

Install Event - Asset Location Sync Details

Destination Name	Local JNDI Name	Remote JNDI Name
MDMInstallEventSyncResponse	jms/LocalMDMInstallEventSyncResponse	jms/ODM-MDM/MDMInstallEventSyncResponse

5. Under the Foreign server, create a Remote Connection Factory.
    - **Name** – Indicates the Name of remote connection factory.
    - **Local JNDI Name** – Add a local JNDI name for the Integration Connection Factory. This JNDI name is added manually later as part of configuration in the WebLogic-ejb-jar.xml -> <weblogic-enterprise-bean> -> <message-driven-descriptor> -> <connection-factory-jndi-name>.
    - **Remote JNDI Name** – JNDI name of the JMS Connection Factory on the Integration SOA server. For example:

Remote Connection Factory Details - MDM

Destination Name	Local JNDI Name	Remote JNDI Name
ODMMDMCF	jms/LocalODMMDMCF	jms/ODM-MDM/ODMMDMCF

3.4.3.2 Configuration File Changes

To configure file changes, you need the following:

- **Configure Message Driven Beans (MDB)**: It is recommended that you use the Oracle Utilities Meter Data Management template and Customer Modification (CM) feature to make changes to these configuration files. This

ensures that your modifications cannot be overwritten by future application patches.

- **Modify files**: ejb-jar.xml and ejb-weblogic-jar.xml
- **Location**: Oracle Utilities Meter Data Management Enterprise Archive (EAR) file.
- **Managing Configuration Files**: Configuration files such as config.xml, ejb-jar.xml and ejb-weblogic-jar.xml are managed through template configuration files which reside in the environment's templates directory. When the initialSetup.sh script is executed, environment specific information is combined with the template to create the target file which is then deployed to the correct location. When the environment is started up (spl.sh start) the changes are automatically deployed to WebLogic.

- **Extending existing templates**: It is possible to extend existing templates with the use of Include template file(s) in the same location as the existing template. Using #ouaf_user_exit within the target template that will be extended, additional configuration from the include template will be processed and appended to the target template where the #ouaf_user_exit is present.
- **Enabling changes for the integration**: To enable your changes for integration with Oracle Utilities Meter Data Management it is recommended that you first make a "CM" copy of the existing template and make your changes to the CM version. If there are any problems with starting the application it is a simple process to delete the CM versions of the files and rerun initialSetup to regenerate and redeploy the original versions.

The following needs to be observed while making configuration file changes:

- The Oracle Utilities Meter Data Management configuration files, ejb-jar.xml and ejb-weblogic-jar.xml, must be modified to configure Message Driven Beans (MDB). MDBs which receive messages from the integration queues. These files are part of the Oracle Utilities Meter Data Management Enterprise Archive (EAR) file.
- The Oracle Meter Data Management application needs to be redeployed after these changes are made.

If you make CM versions of the template files and later install a patch which updates the base template, the CM version will not be updated.

Note: Working example include configuration files are available for download from My Oracle Support in Patch number 14668745 - MDM-ODM INTEGRATION CONFIGURATION. Before installing the examples, please read the Product Fix Design document included in the patch for more information.

To create MDB to receive messages from the Oracle Utilities Meter Data Management inbound queue:

1. Create a new MDB to receive messages from each integration inbound queue. For simplicity, we refer to the names of the target configuration files in the following examples. However, you should make your changes in the templates/cm_<target file>.include version of the file and then execute initalSetup.sh (Unix) or initalSetup.cmd (Windows) to deploy the generated file.
2. Create an MDB for each Oracle Utilities Meter Data Management inbound queue to receive messages and invoke the Oracle Utilities Meter Data Management service.

3. Create or modify cm_ejb-jar.xml.wls.jms_1.include, cm_ejb-jar.xml.wls.jms_2.include, cm_weblogic-ejb-jar.xml.jms.include, cm_config.xml.jms.include, or cm_config.xml.win.jms.include (for Windows systems only) files to configure the MDBs.

   - Add the <message-driven> and <container-transaction> tag for each inbound queue in the ejb-jar.xml.
   - cm_ejb-jar.xml.wls.jms_1.include example:

```
<!--Asset – Device Sync Integration Point -->
<message-driven>
<description>MDB for MDMDeviceSyncRequest</description>
<display-name>MDMDeviceSyncRequest</display-name>
<ejb-name>MDMDeviceSyncRequest</ejb-name>
<ejb-class>com.splwg.ejb.mdb.MessageProcessor</ejb-class>
<messaging-type>javax.jms.MessageListener</messaging-type>
<transaction-type>Bean</transaction-type>
<message-destination-type>javax.jms.Queue</message-destination-type>
</message-driven>

<!--Install Event – Asset Location Sync Integration Point -->
<message-driven>
<description>MDB for MDMInstallEventSyncResponse</description>
<display-name>MDMInstallEventSyncResponse</display-name>
<ejb-name>MDMInstallEventSyncResponse</ejb-name>
<ejb-class>com.splwg.ejb.mdb.MessageProcessor</ejb-class>
<messaging-type>javax.jms.MessageListener</messaging-type>
<transaction-type>Bean</transaction-type>
<message-destination-type>javax.jms.Queue</message-destination-type>
</message-driven>

<!--Contact Sync Integration Point -->
<message-driven>
<description>MDB for MDMContactSyncResponse</description>
<display-name>MDMContactSyncResponse</display-name>
<ejb-name>MDMContactSyncResponse</ejb-name>
<ejb-class>com.splwg.ejb.mdb.MessageProcessor</ejb-class>
<messaging-type>javax.jms.MessageListener</messaging-type>
<transaction-type>Bean</transaction-type>
<message-destination-type>javax.jms.Queue</message-destination-type>
</message-driven>

<!--SP - Location Sync Integration Point -->
<message-driven>
<description>MDB for MDMSPSyncResponse</description>
<display-name>MDMSPSyncResponse</display-name>
<ejb-name>MDMSPSyncResponse</ejb-name>
<ejb-class>com.splwg.ejb.mdb.MessageProcessor</ejb-class>
<messaging-type>javax.jms.MessageListener</messaging-type>
<transaction-type>Bean</transaction-type>
<message-destination-type>javax.jms.Queue</message-destination-type>
</message-driven>
```

cm_ejb-jar.xml.wls.jms_2.include example:

```
<!--Asset – Device Sync Integration Point -->
<container-transaction>
<method>
<ejb-name>MDMDeviceSyncRequest</ejb-name>
<method-name>onMessage</method-name>
</method>
<trans-attribute>NotSupported</trans-attribute>
```

```
</container-transaction>
<!--Contact Sync Integration Point -->
<container-transaction>
<method>
<ejb-name>MDMContactSyncResponse</ejb-name>
<method-name>onMessage</method-name>
</method>
<trans-attribute>NotSupported</trans-attribute>
</container-transaction>

<!--Install Event – Asset Location Sync Integration Point -->
<container-transaction>
<method>
<ejb-name>MDMInstallEventSyncResponse</ejb-name>
<method-name>onMessage</method-name>
</method>
<trans-attribute>NotSupported</trans-attribute>
</container-transaction>

<!--SP - Location Sync Integration Point -->
<container-transaction>
<method>
<ejb-name>MDMSPSyncResponse</ejb-name>
<method-name>onMessage</method-name>
</method>
<trans-attribute>NotSupported</trans-attribute>
</container-transaction>
```

- Modify the cm_weblogic-ejb-jar.xml.jms.include file. Add the <weblogic-enterprise-bean> tag for each inbound queue.
- Add a security role with role cisusers.

The references in <weblogic-enterprise-bean> tag.are as follows:

- <ejb-name> - MDB name given in ejb-jar.xml.
- <destination-jndi-name> - JNDI name provided in JMS module -> Foreign server -> Foreign destination -> Local JNDI name.
- <connection-factory-jndi-name> - JNDI name provided in JMS module -> Foreign server -> Remote Connection Factory -> Local JNDI name.

cm_weblogic-ejb-jar.xml.jms.include example:

```
<!--Asset – Device Sync Integration Point -->
<weblogic-enterprise-bean>
<ejb-name>MDMDeviceSyncRequest</ejb-name>
<message-driven-descriptor>
<pool>
<max-beans-in-free-pool>5</max-beans-in-free-pool>
<initial-beans-in-free-pool>1</initial-beans-in-free-pool>
</pool>
<destination-jndi-name>jms/LocalMDMDeviceSyncRequest</destination-
jndi-name>
<connection-factory-jndi-name>jms/LocalODMMDMCF</connection-factory-
jndi-name>
</message-driven-descriptor>
</weblogic-enterprise-bean>

<!--Contact Sync Integration Point -->
```

```
<weblogic-enterprise-bean>
<ejb-name>MDMContactSyncResponse</ejb-name>
<message-driven-descriptor>
<pool>
<max-beans-in-free-pool>5</max-beans-in-free-pool>
<initial-beans-in-free-pool>1</initial-beans-in-free-pool>
</pool>

<destination-jndi-name>jms/LocalMDMContactSyncResponse</destination-
jndi-name>
<connection-factory-jndi-name>jms/LocalODMMDMCF</connection-factory-
jndi-name>
</message-driven-descriptor>
</weblogic-enterprise-bean>

<!--Install Event – Asset Location Sync Integration Point -->
<weblogic-enterprise-bean>
<ejb-name>MDMInstallEventSyncResponse</ejb-name>
<message-driven-descriptor>
<pool>
<max-beans-in-free-pool>5</max-beans-in-free-pool>
<initial-beans-in-free-pool>1</initial-beans-in-free-pool>
</pool>
<destination-jndi-name>jms/LocalMDMInstallEventSyncResponse</
destination-jndi-name>
<connection-factory-jndi-name>jms/LocalODMMDMCF</connection-factory-
jndi-name>
</message-driven-descriptor>
</weblogic-enterprise-bean>


<!--SP - Location Sync Integration Point -->
<weblogic-enterprise-bean>
<ejb-name>MDMSPSyncResponse</ejb-name>
<message-driven-descriptor>
<pool>
<max-beans-in-free-pool>5</max-beans-in-free-pool>
<initial-beans-in-free-pool>1</initial-beans-in-free-pool>
</pool>
<destination-jndi-name>jms/LocalMDMSPSyncResponse</destination-jndi-
name>
<connection-factory-jndi-name>jms/LocalODMMDMCF</connection-factory-
jndi-name>
</message-driven-descriptor>
</weblogic-enterprise-bean>
```

cm_config.xml.jms.include/ cm_config.xml.win.jms.include example:

```
<jms-system-resource>
<name>ODM2IntegrationExampleModule</name>
<target>myserver</target>
<sub-deployment>
<name>ODM2IntegrationExample</name>
<target>myserver</target>
</sub-deployment>
<descriptor-file-name>jms/Module-for-ODM-integration-example-1-
jms.xml</descriptor-file-name>
</jms-system-resource>
```

3.4.4 XAI Configuration

XAI configuration involves the following:

- XAI JNDI Server
- XAI JMS Queue
- XAI JMS Connection
- XAI Sender
- Outbound Message Type
- External System

3.4.4.1 XAI JNDI Server

Create a new XAI JNDI server **to communicate with the integration layer. It should point** to the Integration SOA server.

To create a XAI JNDI server:

1. In the Admin menu, navigate to the XAI JNDI server.
2. Enter the XAI JNDI server name. For example: **ODM_JNDI**
3. Enter the XAI JNDI server description. For example: MDM-ODM Integration server
4. Enter the Provider URL in the format **t3//<SOA Server>: <SOA Port>**. For example: **t3://soaserver.us.oracle.com:8002**

3.4.4.2 XAI JMS Queue

Create a new XAI JMS queue for each integration queue where Oracle Utilities Meter Data Management sends messages.

To create a XAI JMS queue:

1. In the Admin menu, navigate to XAI JMS queue.
2. Enter the following:
   - **XAI JMS Queue** – Queue name in Oracle Utilities Meter Data Management.
   - **Description** – Queue description
   - **Queue Name** – JNDI name of the queue on the Integration server. For example: jms/ODM-MDM/MDMContactSyncRequest
   - **Target Client Flag** – JMS
   - **XAI JNDI Server** – Select the XAI JNDI server created for integration.

     Note: Only define the queues that Oracle Utilities Meter Data Management will be publishing or writing messages to

The following are few examples to create a XAI JMS queue:

Asset-Device Sync Integration Point Details

XAI JMS Queue	Description	Queue Name	Target Client Flag	XAI JNDI Server
DvcSyncRes	Device Sync Response	jms/ODM-MDM/ MDMDeviceSyncResponse	JMS	ODM_JNDI

SP-Location Sync Integration Point Details

XAI JMS Queue	Description	Queue Name	Target Client Flag	XAI JNDI Server
SPSyncReq	Service Point Sync Request	jms/ODM-MDM/ MDMSPSyncRequest	JMS	ODM_JNDI

Contact Sync Integration Point Details

XAI JMS Queue	Description	Queue Name	Target Client Flag	XAI JNDI Server
ConSyncReq	Contact Sync Request	jms/ODM-MDM/ MDMContactSyncRequest	JMS	ODM_JNDI

Install Event - Asset LocationSync Integration Point Details

XAI JMS Queue	Description	Queue Name	Target Client Flag	XAI JNDI Server
IESyncReq	Install Event Sync Request	jms/ODM-MDM/ MDMInstallEventSyncRequest	JMS	ODM_JNDI

3.4.4.3 XAI JMS Connection

Create a new XAI JMS connection used to connect to the integration queues.

To create a XAI JMS connection:

1. In the Admin menu, navigate to XAI JMS Connection.
2. Enter the following:
   - **XAI JMS Connection** – Connection name in Oracle Utilities Meter Data Management.
   - **Description** – Connection description
   - **XAI JNDI Server** – Select the XAI JNDI server created for this integration (as described in the XAI JNDI Server section).
   - **JNDI ConnectionFactory** – JNDI name of the connection factory on the integration server. For example:

XAI JMS Connection Details

XAI JMS Connection	Description	XAI JNDI Server	JNDI Connection Factory
ODM2_CF	MDM ODM Integration Connection	ODM_JNDI	jms/ODM-MDM/ODMMDMCF

3.4.4.4 XAI Sender

Create a new realtime XAI sender (to communicate with the integration layer) for each Oracle Utilities Meter Data Management Outbound Integration Queue.

To create the XAI sender:

1. In the Admin menu, navigate to XAI Sender.
2. Enter a unique XAI sender and its description.
3. Populate the following values:
   - **XAI Sender** – Sender name in Oracle Utilities Meter Data Management.
   - **Description** – Sender description
   - **Invocation Type** – Real-time
   - **XAI Class** – RTJMSQSNDR (Realtime JMS Queue Sender)
   - **Active** - Select the checkbox.
   - **MSG Encoding** – UTF-8 message encoding
   - **XAI JMS Connection** – XAI JMS connection created for integration.
   - **XAI JMS Queue** – XAI JMS Queue created for the Oracle Meter Data Management outbound queue.
4. Select the Context tab and set values for the following context types:
   - **JMS Message Type** (Bytes(**Y**)/Text(**N**)) – N
   - **JMS User Name** – User for the SOA server to be accessed.
   - **JMS User Password**– Password for the SOA server to be accessed.

     For example:

Asset -Device Sync Integration Point

XAI Sender	Description	XAI JMS Connection	XAI JMS Queue
DvcSyncRes	Device Sync Response Sender to ODM	ODM2_CF	DvcSyncRes

SP-Location Sync Integration Point

XAI Sender	Description	XAI JMS Connection	XAI JMS Queue
SPSyncReq	Service Point Sync Request Sender to ODM	ODM2_CF	SPSyncReq

Contact Sync Integration Point

XAI Sender	Description	XAI JMS Connection	XAI JMS Queue
ContSyncReq	Contact Sync Request Sender to ODM	ODM2_CF	ConSyncReq

Install Event - Asset Location Sync Integration Point

XAI Sender	Description	XAI JMS Connection	XAI JMS Queue
IESyncReq	Install Event Request Sender to ODM	ODM2_CF	IESyncReq

3.4.4.5 Outbound Message Type

Create new Outbound Message Type for each Oracle Utilities Meter Data Management Outbound integration queue.

To create new Outbound Message Type:

1. In the Admin menu, navigate to Outbound Message Type.
2. Enter an outbound message type, its description, and then the detailed description.
3. Select the Outbound Message Business object created for the specific outbound queue.
   For example:

Outbound Message Type for Asset -Device Sync

Outbound Message Type	Description	Business Object
DM-OMT-DEF	Device Sync Response Outbound Message Type	D1-OngoingSyncReqAckMsg

Outbound Message Type for SP-Location Configuration Sync

Outbound Message Type	Description	Business Object
DMSPODMMSG	ODM Service Point Outbound Message	D1-OutboundMessage

Outbound Message Type for Contact Sync

Outbound Message Type	Description	Business Object
DM-CONTMSG	ODM Contact Outbound Message	D1-OutboundMessage

Outbound Message Type for Install Event - Asset Location Sync

Outbound Message Type	Description	Business Object
DMIERESMSG	ODM Install Event Outbound Message	D1-OutboundMessage

3.4.4.6 External System

Create a new external system to support the integration by performing the below steps:

To create a new external system:

1. In the Admin menu, define an external system.
2. Enter a unique name for the external system and its description.
   For example: Name = CI_ODM, Description = Oracle Utilities Operational Device Management
3. Set the Our Name in Their System field to Meter Data Management.
4. Associate the outbound message types created to the external system. For each outbound message type, set the following:
   - **Outbound Message Type** – Set the outbound message type created for Oracle Utilities Meter Data Management outbound queue.
   - **Processing Method** – Real-time
   - **XAI Sender** – Set the XAI sender created for the queue.
   - **Message XSL**-For example: External System – ODM

External System Details

Outbound Message Type	Processing Method	XAI Sender	Message XSL
DM-OMT-DEF	Real-time	DvcSyncRes	D1-MDMODMResponseAddNamespace.xsl
DMSPODMMSG	Real-time	SPSyncReq	D1-MDMODMRequestAddNamespace.xsl
DM-CONTMSG	Real-time	ContSyncReq	D1-MDMODMRequestAddNamespace.xsl
DMIERESMSG	Real-time	IESyncReq	D1-MDMODMRequestAddNamespace.xsl

For more information about configuration guidelines, see the *Oracle Utilities Meter Data Management User Documentation*.

3.5 Setting up the Process Integration

The following sections describe how to configure integration pack to meet the requirements for two-way integration.

Configuration steps include setting the following:

- Setting Configuration Properties
- Setting System Properties
- Domain Value Maps
- Error Handling

3.5.1 Setting Configuration Properties

The ConfigurationProperties.XML file contains properties which can be defaulted in the integration. Also, it contains flags to enable extension points within the integration.

ConfigurationProperties.XML is located in MDS under the directory apps/ODM-MDM/AIAMetaData/config.

> **Note**: Whenever the ConfigurationProperties.XML file is updated, it must be reloaded to MDS for updates to be reflected in the applications or services that use the updated properties. You can perform the reload by rebooting the SOA server.

3.5.2 Setting System Properties

There are two sets of configuration properties described in this section:

- Module Configurations are the properties shared by multiple integration flows within this integration.
- Service Configurations are the properties used by a specific BPEL process.

3.5.2.1 Module Configurations

The Module Configuration has application level properties and which are used by all the SOA composites.

Module Configuration Details

Module Name	Default / Shipped Value	Description
ODM.Generic.MessageCategory	11017	Message category that the integration layer uses for ODM error messages
ODM.GenericBusinessException.Message Number	11001	Message number that the integration layer uses for generic ODM errors
ODM.MessageCategoryNumber.Separator		Separator used to separate the ODM generic message category and generic business exception message number when setting the value in the DVM

Module Name	Default / Shipped Value	Description
MDM.Generic.MessageCategory	11017	Message category that the integration layer uses for Oracle Utilities Meter Data Management error messages
MDM.GenericBusinessException.Message Number	11001	Message number that the integration layer uses for generic Oracle Utilities Meter Data Management errors
MDM.MessageCategoryNumber.Separator		Separator used to separate the Oracle Utilities Meter Data Management generic message category and generic business exception message number when setting the value in the DVM
SOA-INFRA.AuditLevel	ON	This property needs to be set to OFF if the Audit Level is set to OFF for the BPEL processes. If the setting is OFF, then error handling does not use the composite and component instance IDs to log the error message.
ErrorHandling.GenericEmailID		This property is used to set the administrator email ID for the errorhandling process to send out an email in case of a critical failure where even the Errorhandling process fails.

3.5.2.2 Service Configurations

The Service Configuration properties are specific to SOA composites. These are used to make changes in specific composite behavior.

Service Configuration details are as explained in the table below:

Service Configuration Details

Service Name	Property Name	Default / Shipped Value	Description
MDMODMContactSyncReq EBF	Extension.PreXform MDMToODM	false	If set to true, the pre transformation extension service is invoked.
	Extension.PostXform MDMToODM	false	If set to true, the post transformation extension service is invoked.
	BusinessError.Notific ationFlag	true	If set to true, business error notification is sent via Email.

Service Name	Property Name	Default / Shipped Value	Description
	TechnicalError.NotificationFlag	true	If set to true, technical error notification is sent via Email.
ODMMDMContactSyncRespEBF	Extension.PreXformODMToMDM	false	If set to true, the pre transformation extension service is invoked.
	Extension.PostXformODMToMDM	false	If set to true, the post transformation extension service is invoked.
	BusinessError.NotificationFlag	true	If set to true, business error notification is sent via Email.
	TechnicalError.NotificationFlag	true	If set to true, technical error notification is sent via Email.
MDMODMSPNodeSyncReqEBF	Extension.PreXformMDMToODM	false	If set to true, the pre transformation extension service is invoked.
	Extension.PostXformMDMToODM	false	If set to true, the post transformation extension service is invoked.
	BusinessError.NotificationFlag	true	If set to true, business error notification is sent via Email.
	TechnicalError.NotificationFlag	true	If set to true, technical error notification is sent via Email.
ODMMDMSPNodeSyncRespEBF	Extension.PreXformODMToMDM	false	If set to true, the pre transformation extension service is invoked.
	Extension.PostXformODMToMDM	false	If set to true, the post transformation extension service is invoked.
	BusinessError.NotificationFlag	true	f set to true, business error notification is sent via Email.
	TechnicalError.NotificationFlag	true	If set to true, technical error notification is sent via Email.

Service Name	Property Name	Default / Shipped Value	Description
MDMODMInstallEventAssetNodeSyncReqEBF	Extension.PreXformMDMToODM	false	If set to true, the pre transformation extension service is invoked.
	Extension.PostXformMDMToODM	false	If set to true, the post transformation extension service is invoked.
	BusinessError.NotificationFlag	true	If set to true, business error notification is sent via Email.
	TechnicalError.NotificationFlag	true	If set to true, technical error notification is sent via Email.
	Customize.AssetMovement.flag	false	This flag if false will indicate that the asset movements group is a pass through for integration. If this flag is set to true implementation can customize the values being set in the asset movement group in the integration layer
ODMMDMInstallEventAssetNodeSyncRespEBF	Extension.PreXformODMToMDM	false	If set to true, the pre transformation extension service is invoked.
	Extension.PostXformODMToMDM	false	If set to true, the post transformation extension service is invoked.
	BusinessError.NotificationFlag	true	If set to true, business error notification is sent via Email.
	TechnicalError.NotificationFlag	true	If set to true, technical error notification is sent via Email.
ODMMDMAssetDeviceSyncReqEBF	Extension.PreXformODMToMDM	false	If set to true, the pre transformation extension service is invoked.
	Extension.PostXformODMToMDM	false	If set to true, the post transformation extension service is invoked.
	BusinessError.NotificationFlag	true	If set to true, business error notification is sent via Email.

Service Name	Property Name	Default / Shipped Value	Description
	TechnicalError.NotificationFlag	true	If set to true, technical error notification is sent via Email.
MDMODMAssetDeviceSyncRespEBF	Extension.PreXformMDMToODM	false	If set to true, the pre transformation extension service is invoked.
	Extension.PostXformMDMToODM	false	If set to true, the post transformation extension service is invoked.
	BusinessError.NotificationFlag	true	If set to true, business error notification is sent via Email.
	TechnicalError.NotificationFlag	true	If set to true, technical error notification is sent via Email.

3.5.3 Domain Value Maps

Domain value maps (DVMs) are a standard feature of the Oracle SOA Suite which maps codes and other static values across applications. For example: "US" and "USA"

DVMs are static in nature, though Administrators can add additional maps as needed. Transactional business processes never update DVMs - they only read from them. They are stored in XML files and cached in memory at runtime.

To maintain information within the domain value maps:

1. Open a browser and access the SOA Composer application (http://host:port/soa/composer/)
2. On the SOA Composer, click the "Open" dropdown and select "Open DVM". This displays a list of all DVM files in MDS.
3. Select the relevant DVM you wish to maintain.
4. Edit the selected DVM. The Edit button in the top navigation bar enables editing the DVM.
5. Once the DVM has been edited, click Save in the navigation bar. This saves the DVM data for that session.
6. Click Commit after updating each DVM. This saves the DVM data in MDS.

The DVMs for the integration are listed as follows:

Domain Value Code Details for the Integration Points

DVM	Integration Points	Description
MDM_ODM_ErrorCode	All	Used by the integration code to transform Oracle Utilities Meter Data Management Message Category and Message number to Oracle Utilities Operational Device Management Message Category and message number and vice versa. This helps the implementation layer to map specific message numbers to specific error codes, and thus provide more user-friendly error messages in Oracle Utilities Operational Device Management and Oracle Utilities Meter Data Management.
ODM_MDM_MO	Asset – Device Sync	Maps Oracle Utilities Operational Device Management Maintenance Object value to the corresponding Oracle Utilities Meter Data Management Maintenance Object value.
MDM_ODM_MO	Contact Sync , SP – Location Sync and Install Event – Asset Location Sync	Maps Oracle Utilities Meter Data Management Maintenance Object value to corresponding Oracle Utilities Operational Device Management Maintenance Object value.
MDM_ODM_ContactType	Contact Sync	Maps Oracle Utilities Meter Data Management Contact type value to corresponding Oracle Utilities Operational Device Management Contact Type value.
MDM_ODM_AssetDeviceType	Asset – Device Sync	Maps Oracle Utilities Operational Device Management asset type to corresponding Oracle Utilities Meter Data Management
MDM_ODM_AssetDeviceStatus	Asset – Device Sync	Maps Oracle Utilities Operational Device Management asset status to corresponding Oracle Utilities Meter Data Management
MDM_ODM_SPNodeType	SP – Location Sync	Maps Oracle Utilities Meter Data Management service point type to corresponding Oracle Utilities Operational Device Management node

DVM	Integration Points	Description
MDM_ODM_NodeDisposition	SP – Location Sync	Maps Oracle Utilities Meter Data Management Measurement node disposition to corresponding Oracle Utilities Operational
MDM_ODM_DisconnectLocation	SP – Location Sync	Maps Oracle Utilities Meter Data Management disconnect location code to corresponding Oracle Utilities Operational Device Management disconnect
MDM_ODM_OkToEnterCode	SP – Location Sync	Maps Oracle Utilities Meter Data Management ok to enter code to corresponding Oracle
MDM_ODM_Country	SP – Location Sync	Maps Oracle Utilities Meter Data Management country code to corresponding Oracle Utilities Operational Device Management country code.
MDM_ODM_LifeSupportSensitiveLoad	SP – Location Sync	Maps Oracle Utilities Meter Data Management life support sensitive load flag to corresponding Oracle Utilities
MDM_ODM_TimeZone	SP – Location Sync	Maps Oracle Utilities Meter Data Management time zone code to corresponding Oracle Utilities Operational Device Management time zone code. This is a valid time zone code

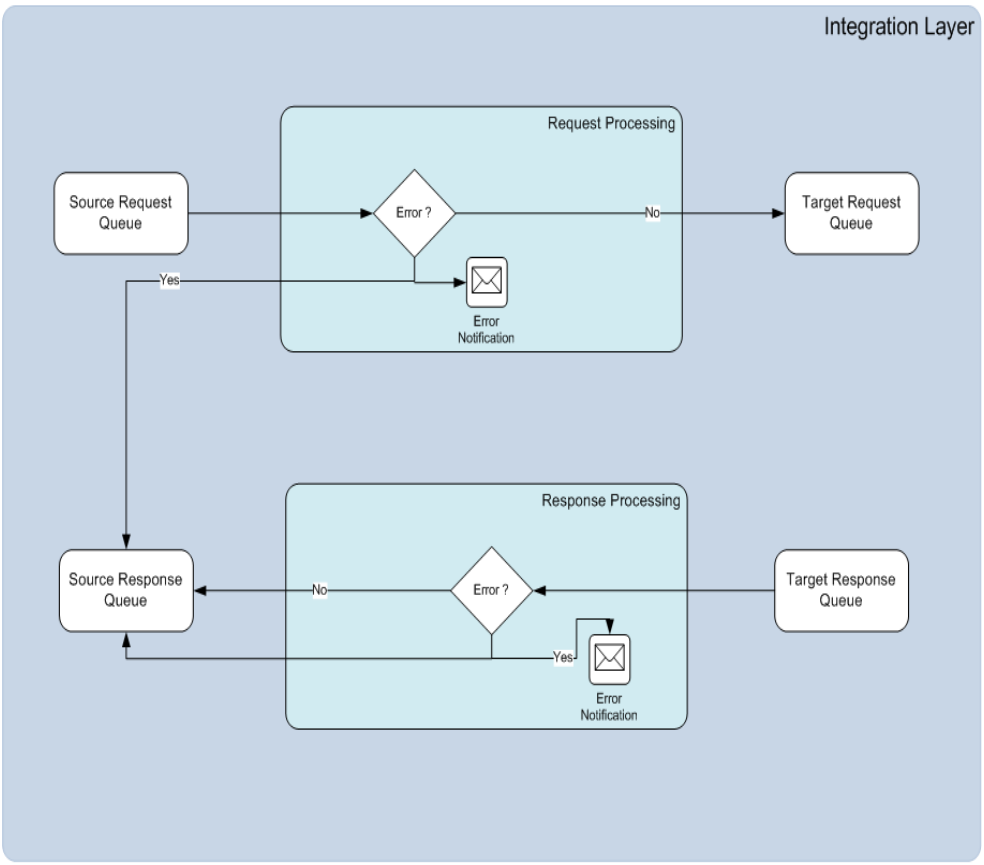
For more information about Domain Value Maps, refer the chapters *Working with Domain Value Maps* and *Using SOA Composer with Domain Value Maps* in Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite.

3.5.4 Error Handling

The integration includes two types of errors:

- **Business Errors**– Triggered when there is a transformation error in the integration layer. Business errors are sent back to the source application and can be re-tried from there.
- **Technical Errors** – Triggered when there are connectivity issues between queues. Technical errors are sent to the error queue and can be re-tried from integration layer.

The following diagram depicts the process for error handling:



Error Handling

Integration Error Details

S. No	Integration Flow	Type of error	Action	Notification Type	Retry
A1	Master Data Sync – ODM originated request processing (e.g.. Asset-Device Sync Request flow)	Business error	Message is sent to ODM Response Queue (e.g. ODM Asset Response Queue)	Email (optional) and ODM ToDo	Data correction in ODM
A2		Technical error	Message is rolled back to ODM Request Error Queue (e.g. ODM Asset Request Error Queue)	Email (optional)	Administrator has to move the messages to ODM Request Queue from WebLogic Admin console. (e.g. ODM Asset Request Queue)
A3	Master Data Sync – MDM originated response processing (e.g.. Asset - Device Response Sync flow)	Business error	Message is sent to ODM Response Queue. . (e.g. ODM Asset Response queue)	Email (optional) and ODM ToDo	Data correction in ODM
A4		Technical error	Message is rolled back to the MDM Response Error Queue (e.g. MDM Device Response Error Queue)	Email (optional)	Administrator has to move the messages to MDM Response Queue from WebLogic Admin console. (e.g. MDM Device Response Queue)
B1	Master Data Sync - MDM originated request processing (e.g..Contact Sync Request flow)	Business error	Message is sent to MDM Response Queue. (e.g. MDM Contact Response Queue)	Email (optional) and MDM ToDo	Data correction in MDM

S. No	Integration Flow	Type of error	Action	Notification Type	Retry
B2		Technical error	Message is rolled back to MDM Request Error Queue. (e.g. MDM Contact Request Error Queue)	Email (optional)	Administrator has to move the messages to MDM Request Queue from WebLogic Admin console. (e.g. MDM Contact Request Queue)
B3	Master Data Sync - ODM originated response processing (e.g.Contact Sync Response flow)	Business error	Message is sent to MDM Response Queue. (e.g. MDM Contact Response queue	Email (optional) and MDM ToDo	Data correction in MDM
B4		Technical error	Message is rolled back to ODM Response Error Queue. (e.g. ODM Contact Response Error Queue)	Email (optional)	Administrator has to move the messages to ODM Response Queue from WebLogic Admin console. (e.g. MDM Contact Response Queue)

To retry the technical error failure messages:

1. In the WebLogic console, navigate to **Services > Messaging > JMS Modules**.
2. Select ODM-MDM Integration JMS Module to display all queues related to this integration.
3. Select the appropriate error queue and click the **Monitoring** tab.
   This tab displays the details about messages in the queue in a table.
4. Select the checkbox in the details table and click **Show Messages**.
   This displays all the messages in the error queue.
5. Click **Move** and select **Move All**.
6. Select the ODM-MDM JMS server to move messages and then click **Next**.
7. Select the correct parent queue for the error queue from the drop-down menu and click **Finish**.

This action moves all messages to the source queue, so that the integration layer processes all messages again.

3.5.4.1 Error Notification Setup

Error Notification setup to enable EMAIL notification for ODM-MDM Error Handling module can be done by following the steps shown below:

To enable EMAIL notification for ODM-MDM Error Handling module:

1. Login to the Enterprise Manager console.
2. Expand **SOA** and then right-click **SOA Infra**. From the menu, click **SOA Administration** and then click **Workflow Notification Properties**.
3. From the drop-down list, select **EMAIL**.
4. Enter the Email IDs in the From address field.

3.5.4.2 INTEGRATION_ERR_STORE

The INTEGRATION_ERR_STORE table is used to store all the error details for each message failure. The table is populated for each integration point based on the BusinessError.NotificationFlag and TechnicalError.NotificationFlag properties for each service in the ConfigurationProperties.xml file.

3.5.4.3 INTEGRATION_ERR_LOOKUP

The error handling module configuration is governed by the Integration_err_lookup table. This table contains processing instructions for each composite. The Error_Processing_Parent composite picks data for one composite and calls error_Processing_Detail for processing. The configuration in this table is used to process the error records stored in the INTEGRATION_ERR_STORE table.

Integration Err_Lookup Details

S. No.	Column Name	Description	Default/ Suggested values
1	LookUp_ID	Sequence ID of entry in this table. This is auto generated.	Auto generated
2	IP_Name	Name of the composite processed. Example: OUMDMOUODMReplReadReqEBF	This column is prepopulated with the individual enterprise business flow name. Do not modify. Modifying this value will break the code.
3	Processing_Status	Current status of processing it has to be one of the following: 1. HALTED (waiting for manual intervention), 2. NOT REQUIRED 3. ALIVE	NOT REQUIRED
4	Run_Flag	Processing flag status, Y or N. Unread value = N, read value =Y	N
5	Next_Runtime	Next runtime when the error record should be processed for this composite.	SYSDATE+200
6	Halt_For_Error	Allowed values **Y** or **N**. When set to Y, manual intervention is required after one successful error record processing. When set to N, processing continues without halting.	N

S. No.	Column Name	Description	Default/ Suggested values
7	RunTime_Interval	Runtime in minutes after which the next error processing should be done. Example : P10Y0M0DT0H0M0S Next processing is done after 10 years 0 months 0 days 0 hours 0 minutes and 0 seconds This value must be updated based on the business requirement. Setting fewer intervals may have impact on performance.	Default : P10Y0M0DT0H0M0S
8	Email_ID	Email ID where error notifications are sent. This value can be different or same for all the composites.	Default : email@email.com
9	Email_Content_Type	GENERIC – One Email is sent for all errors. No detail information is included. SINGLE – One Email is sent for all errors with details included in the attachment. MULTIPLE – Multiple Emails are sent and each email has information equal to the value specified in Error_Count_Per_Notification column. Values are case sensitive and must always be given in upper case.	Default : GENERIC

S. No.	Column Name	Description	Default/ Suggested values
10	Email_XSL	XSL to be applied for creating Email Content which includes subject/body and attachment. Look and feel can be modified here.	Default file is provided for all the composites and present under the xsl folder of composite. Example: xsl/ Transformation_Create_Email.xsl Copy this to the mds folder and enter the mds path in this column for additional configuration.
11	Error_Count_Per_Processing	A notification is sent after the number of records set here is processed. For example: If this is set to 50, then an email notification containing 50 reocrds is sent after 50 records are created in the error store.	Default : 100
12	Email_Attachment_Location	Location where the Email attachment is created on the server. This value should point to the location/ folder where the attachment should be stored. This is used to create the attachment file in the following format. INTEGRATION_ERR_LOOKUP.Email_Attachment_Location + IP_Name + Date (in YYYYMMDDHH24MMSS)	
13	Email_Attachment_Flag	**XAI JMS Queue**	**Description**
14	Publish_Human_Task_Flag	SPSyncReq	Service Point Sync Request

S. No.	Column Name	Description	Default/ Suggested values
15	ID_Human_Task	User/ Group ID to which human task should be published in case Halt_For_Error is set to Y. This ID must be present in the WebLogic realm pointed by fusion middleware.	weblogic
16	Last_Updated_Date	Last updated date time	SYSDATE
17	Purge_Error_Store_Flag	**Y** – Purge data **N** – No purge required The process PurgeIntegrationErrorStore is deployed when the flag, purge.process.deploy= true (in the deploy.properties file) is set to true during installation. If flag.purge.process = false, then value of this column Purge_Error_Store_Flag will always be N.	Default : N
18	Purge_Processing_Status_Flag	Y – Purge Processing in process N – Purge processing not happening The process PurgeIntegrationErrorStore is only deployed when the flag, purge.process.deploy= true (in the deploy.properties file) is set to true during installation. If flag.purge.process = false then value of this column Purge_Error_Store_Flag will always be N.	Default : N

S. No.	Column Name	Description	Default/ Suggested values
19	Purge_Frequency	No of days after which data should be purged. This will be in picture format Example : P10Y0M0DT0H0M0S Next processing will be done after 10 years 0 months 0 days 0 hours 0 minutes and 0 seconds. This value has to be updated based on the business requirement. Setting fewer intervals may have impact on performance. Need to set this value appropriately. Applicable only when flag.purge.process = true in deploy.properties file during installation and the process PurgeIntegrationError Store ID deployed.	Default : P10Y0M0DT0H0M0S
20	Next_Purge_Date	Next purge date. Format: Next_Purge_date + Purge_Frequency Applicable only when flag.purge.process = true in deploy.properties file during installation and the process PurgeIntegrationError Store ID deployed.	SYSDATE+100

S. No.	Column Name	Description	Default/ Suggested values
21	Purge_File_Name	Directory name where the purge file should be stored. Applicable only when flag.purge.process = true in deploy.properties file during installation and the process PurgeIntegrationError Store ID deployed.	'location on server where purge record should be persisted'

To customize error email notifications for individual integration points:

1. Use the composite: UpdateIntegrationErrorLookupTable.
2. Open the following URL in a browser to get the screen that provides options to update the contents of table. http://<hostname>:<soa server port>/soa-infra/services/ODM-MDM/ UpdateIntegrationErrorLookupTable/updateintegrationerrorlookuptablebpel_client_ep?
3. Expand WS-Security and provide authentication information.
   Use the username and password that you used to log on to the WebLogic Enterprise Manager console.
4. Expand the paylod section. This displays several editable text fields. Only the ipName field is mandatory and should be entered as one of the values from INTEGRATION_ERR_LOOKUP.IP_NAME field.
   By default all the checkboxes appearing next to the text fields are checked.
5. Provide values in the text field. If you do not want to have a particular value updated, then uncheck the box.

Chapter 4

Monitoring and Troubleshooting

This section discusses in details the following:

- Monitoring from Oracle Utilities Operational Device Management
- Monitoring from Oracle Utilities Meter Data Management
- Monitoring from the Integration layer
- Troubleshooting

4.1 Monitoring from Oracle Utilities Operational Device Management

This section describes in detail the following:

- Oracle Utilities Operational Device Management Error Logs
- Notifications
- Connection Errors

4.1.1 Oracle Utilities Operational Device Management Error Logs

The following error logs can be monitored for Oracle Utilities Operational Device Management:

- Errors related to the online integration invocation from Oracle Utilities Operational Device Management (OUODM) are stored in the ODM_ENVIRONMENT_NAME/logs/system folder.

  For example: V231_ODM_PERF_BLD10_LIN_ORA_WLS/logs/system\

- Errors related to batch integration invocation from ODM are stored in the $SPLOUTPUT/ ODM_ENVIRONMENT_NAME folder.

  For example: /spl/sploutput/V231_ODM_PERF_BLD10_LIN_ORA_WLS

For more information about errors and notifications, see the Oracle Utilities Operational Device Management documentation.

4.1.2 Notifications

When Oracle Utilities Operational Device Management sends a request message out to Oracle Meter Data Management, it expects a response back. It can get a positive response when the message is processed successfully or can get an error response when a business error is encountered in the integration or from the target application.

When Oracle Utilities Operational Device Management receives the response message from the inbound Oracle Utilities Operational Device Management response queue, the message is parsed and converted to an XML document, checked that the XML is valid and that the XML has a valid XAI inbound service.

- If there is an error encountered while processing the message, EJBException will be thrown causing the message to be rolled back to the corresponding Oracle Utilities Operational Device Management response error queue and a To Do entry is created, if configured.

  For example: If Oracle Utilities Operational Device Management receives a contact sync response message from the Oracle Utilities Operational Device Management contact sync response queue and an error is encountered, the message will be moved to the ODM contact sync response error queue.

- If the message was processed successfully, the Business Object or Business Service or Service Script (BO/BS/SS) defined on XAI Inbound Service is invoked. If an application error is encountered inside the BO/BS/SS processing, the message will not be rolled back to the error queue. Only a To Do entry is created, if configured. Otherwise, the error will only be seen in the spl-service.log file.

  Note: The XAI inbound service is invoked to process the response message. Regardless of whether To-Do was set up or not, the errors are logged in spl-service.log file.

4.1.2.1 Setup To Do Entry for JMS message error

The following setup To Do Entry for JMS message error are required:

- XAI Options
- XAI Inbound Services

XAI Options

Define To Do Type for Inbound JMS Message Errors XAI Option. Use To Do Type F1-INJMS (Inbound JMS Message In Error). This To Do Type is delivered with the application. Implementation can define a custom To Do Type, if needed.

XAI Inbound Services

For every XAI Inbound Service used to process the different Sync Response, Billing Determinant Response and Replacement Reads Request, the **Post Error** checkbox must be set to **Yes**.

4.1.3 Connection Errors

Information about errors can be found in log files. For information about error logs and their respective folders, see the section Oracle Utilities Operational Device Management Error Logs.

4.2 Monitoring from Oracle Utilities Meter Data Management

This section describes in detail the following:

- Oracle Utilities Meter Data Management Error Logs
- Notifications
- Connection Errors

4.2.1 Oracle Utilities Meter Data Management Error Logs

Following error logs for Oracle Utilities Meter Data Management can be monitored:

- Errors related to the online integration invocation from Oracle Utilities Meter Data Management are stored in the MDM_ENVIRONMENT_NAME/system/logs folder.

  For example: V201_MDM_BLD10_LIN_ORA_WLS/logs/system

- Errors related to batch integration invocation from Oracle Utilities Meter Data Management are stored in the $SPLOUTPUT/MDM_ENVIRONMENT_NAME folder.

  For example: /spl/sploutput/V201_MDM_BLD10_LIN_ORA_WLS

For more information about errors and notifications, refer to the Oracle Utilities Meter Data Management documentation.

4.2.2 Notifications

When Oracle Utilities Meter Data Management receives a request message from Oracle Utilities Operational Device Management, it will send a response back to Oracle Utilities Operational Device Management. It can send a positive response when the message is processed successfully or can send an error response when a business error is encountered.

When Oracle Utilities Meter Data Management receives the request message from the inbound MDM request queue, the message is parsed and converted to an XML document, checked that the XML is valid and check that the XML has a valid XAI inbound service.

- If there is an error encountered while processing the message, EJBException will be thrown causing the message to be rolled back to the corresponding Oracle Utilities Meter Data Management request error queue and a To Do entry is created, if configured.

  For example: If Oracle Utilities Meter Data Management receives a contact sync request message from the Oracle Utilities Meter Data Management contact sync request queue and an error is encountered, the message will be moved to the Oracle Utilities Meter Data Management contact sync request error queue.)

- If the message was processed successfully, the Business Object or Business Service or Service Script (BO/BS/SS) defined on XAI Inbound Service is invoked. If an application error is encountered inside the BO/BS/SS processing, the message will not be rolled back to the error queue. Only a To Do entry is created, if configured. Otherwise, the error will only be seen in the spl-service.log file.

Note: The XAI inbound service is invoked to process the request message. Regardless of whether To-Do was set up or not, the errors are logged in spl-service.log file.

4.2.2.1 Setup To Do Entry for JMS message Error

Following setup for To Do Entry for JMS message error is required:

- XAI Options
- XAI Inbound Service

XAI Options

Define To Do Type for Inbound JMS Message Errors XAI Option. Use To Do Type F1-INJMS (Inbound JMS Message In Error). This To Do Type is delivered with the application. Implementation can define a custom To Do Type if needed.

XAI Inbound Service

For every XAI Inbound Service used to process the different Sync Request and Billing Determinant Request, the **Post Error** checkbox must be set to **Yes**.

4.2.3 Connection Errors

Information about errors can be found in log files. For information about error logs and their respective folders, see the section Oracle Utilities Meter Data Management Error Logs.

4.3 Monitoring from the Integration layer

To monitor the integration flow using the Integration, use any of the following:

- Monitoring the Composite Instances Using WebLogic SOA Enterprise Manager
- Monitoring the WebLogic Logs
- Monitoring the Queues Using WebLogic Console
- Data Purge

4.3.1 Monitoring the Composite Instances Using WebLogic SOA Enterprise Manager

Perform the following steps to monitor using the WebLogic SOA Enterprise Manager:

To monitor the Composite Instances using the WebLogic SOA Enterprise Manager:

1. Log in to the WebLogic SOA Server Enterprise Manager, and then navigate to **SOA > SOA-Infra > ODM-MDM**.
   All composite processes deployed for integration are available under the partition ODM-MDM.
2. Select the appropriate process to list all the instances for the processes sorted by time of execution.
   The instances also have the request ID as part of the display name.
3. Click the appropriate process instance and it will display the flow for the process.
   The composite flow lists all activities in the process instance.

4.3.2 Monitoring the WebLogic Logs

Log in to the machine where SOA server is installed. The SOA logs are stored in: <WebLogic installation folder>/user_projects/domains/<SOA Domain name>/servers/<SOA Server name>/logs

For example: /slot/ems1234/oracle/Middleware/user_projects/domains/soa_domain/servers/soa_server1/logs

4.3.3 Monitoring the Queues Using WebLogic Console

Perform the following steps to monitor the queues using the WebLogic Console:

To monitor the queues from the WebLogic Console:

1. Log in to the WebLogic Console, and then go to the **Services > Messaging > JMS Modules**.
   All queues used for the integration are available in the JMS Module **ODM2MDM2FJM**.
2. Select the appropriate queue on the list and navigate to the Monitoring tab. In the Monitor tab, the user can check if the messages are stuck in the queue, if there are no consumers listening to the queue, and how many consumers are listening to the queue. If the Consumers Current column is 0, it means no consumers are listening to the queue.
3. To check the message rolled back to the error queue, select the appropriate error queue on the list and navigate to the Monitoring tab. In the Monitor tab, the user can see the message.

4.3.4 Data Purge

To maintain maximum system integrity, the Oracle Fusion Middleware database should be purged periodically.

For information about how to complete this task, refer to the note 815896.1 on https://support.oracle.com

4.4 Troubleshooting

At times, the integration might experience errors or issues with connection, processing, or sending or receiving messages. Following are the common scenarios which help you to troubleshoot error, if any, and find possible solutions.

> **Note**: The source application is the one sending out the message and the target application is the one receiving the message.
> Example: ODM is sending a Asset sync request message to MDM, ODM is the source application and MDM is the target application. The source queue is ODMAssetSyncRequest and the target queue is MDMDeviceSyncRequest. The source error queue is ODMAssetSyncRequestError and the target queue is MDMDeviceSyncRequestError.

Error 1: Source application sends out a message but the message does not reach the source queue

To troubleshoot the above error, go to WebLogic Console to check if the message reached the source queue.

Refer to Monitoring the Queues Using WebLogic Console section for more information

To resolve this error, do the following:

1. Check the source application logs to see if any errors are encountered while trying to send the message out.
   Refer to Oracle Utilities Meter Data Management Error Logs or Oracle Utilities Operational Device Management Error Logs sections for more information on where to find the logs
2. Check the source application's XAI Configuration to ensure they are configured correctly.
   Refer to the Setting up Oracle Utilities Operational Device Management– XAI Configuration or Setting Up Oracle Utilities Meter Data Management – XAI Configuration sections for more information.

Error 2: Source application sends out a message but the message does not reach the target Queue

To resolve this error, perform the following:

1. Check if the BPEL processes are running.
   Refer to the Monitoring the Composite Instances Using WebLogic SOA Enterprise Manager section for more information.
   - If WebLogic SOA Enterprise Manager is not accessible or the BPEL processes cannot be seen found in the Weblogic SOA Enterprise Manger, restart the SOA managed server. Refer to the Oracle Utilities Integration for Device Operations Installation Guide, under Restarting SOA Managed Server or it can also be started from the WebLogic console.
   - If WebLogic SOA Enterprise Manager is accessible but the BPEL process is not active, activate or start up the process from the WebLogic SOA Enterprise Manager.
2. If the BPEL processes are running, check if the message has faulted or encountered an error.
   Refer to the Monitoring the Composite Instances Using WebLogic SOA Enterprise Manager section for more information.
   - If a technical error is encountered, the message is rolled back to the corresponding source error queue. Fix the error and move the message back to the source queue to retry.
   - From WebLogic SOA Enterprise Manager, check the appropriate process instance flow trace to see the error details.
   - Check the logs.
     Refer toMonitoring the WebLogic Logs section for more information.

Error 3: Source application sends out a message, message successfully processed by the integration, but the message does not reach the target application

To resolve this error, perform the following:

1. In Weblogic SOA Enterprise Manager, check the process to see if the message was successfully processed by the integration layer.
   Refer to the Monitoring the Composite Instances Using WebLogic SOA Enterprise Managersection for more information.
2. If a successful instance of the message was found in the WebLogic SOA Enterprise Manager, check the target queue to see if the message exists in the queue.

- Check the corresponding target queue of the process to see if there is a current or pending message stuck in the queue. The possible cause is that no consumers are listening to the target queue.

  Refer to the Monitoring the Queues Using WebLogic Console section for more information.
  To fix this, restart the target application.

- If there are still no consumers listening to the target queue after bouncing the application, check the target application's JMS Configuration to make sure they are configured correctly.

  Refer to Setting up Oracle Utilities Operational Device Management – JMS Configuration or Setting Up Oracle Utilities Meter Data Management – JMS Configuration for more information.

  After changing the JMS configuration of the target application, restart the target application.

- Check the source application logs to see if any errors are encountered while trying to send the message out.
  Refer to the Oracle Utilities Operational Device Management Error Logs or Oracle Utilities Meter Data Management Error Logs section for more information on where to find the logs.

3. If no message is stuck in the target queue, check the target application logs to see if any errors are encountered while trying to processe the message received.
   Refer to the Oracle Utilities Operational Device Management Error Logs or Oracle Utilities Meter Data Management Error Logs section for more information on where to find the logs.

Chapter 5

Customization Options

This chapter discusses the following customization options available for the integration layer:

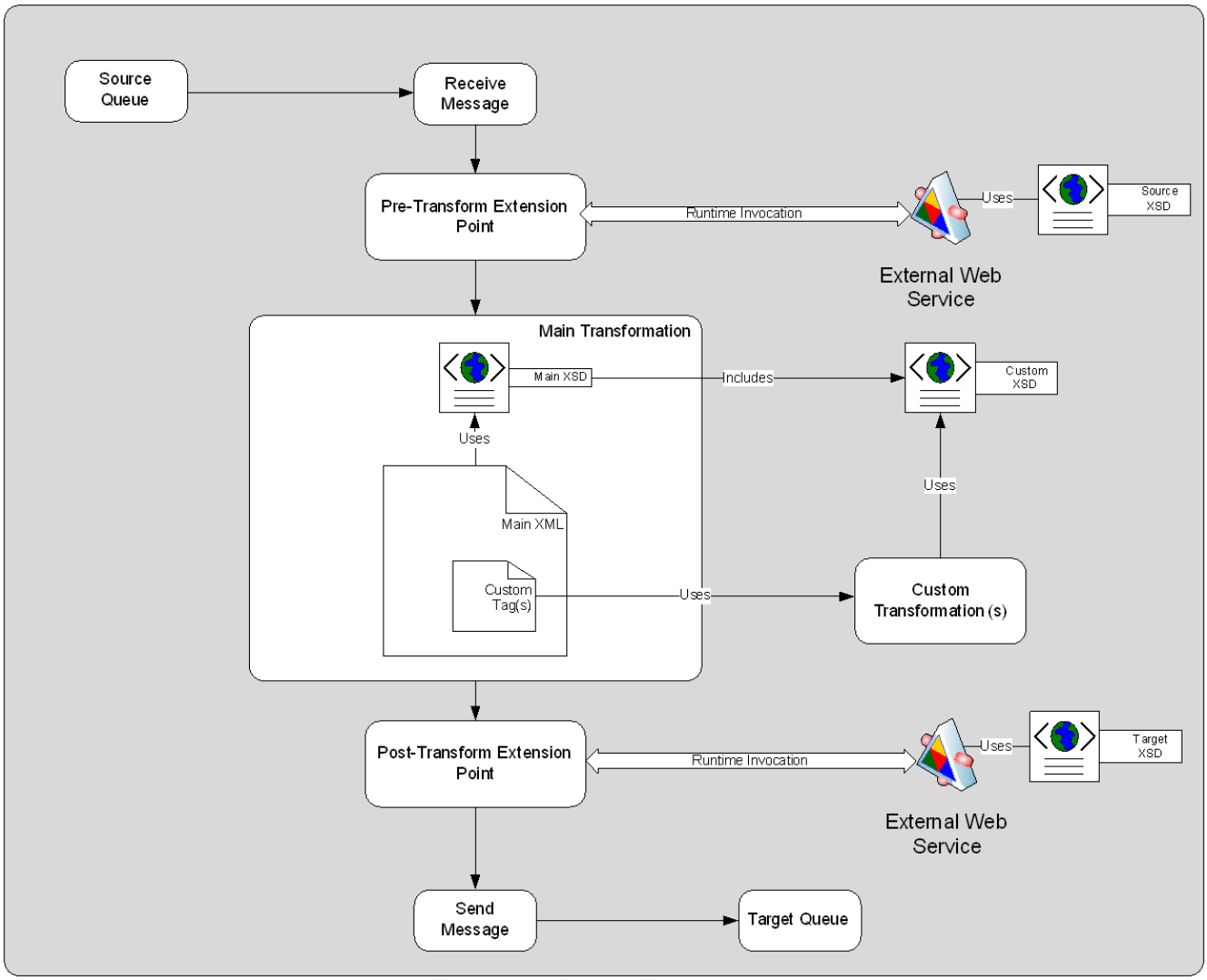
- Extension Methods

5.1 Extension Methods

The integration process allows extensibility of transaction messages using the following methods:

- Pre transformation Extension Point
- Post transformation Extension Point
- Custom Transformations
- Steps to Implement Extension Points
- Steps to Implement Custom Transformations

The following diagram provides a graphical representation of this functionality:



Integration Process Extensibility

5.1.1 Pre transformation Extension Point

The pre transformation extension point is invoked before the main transformation is executed. This transformation aids in transforming the source XML coming as an input to the integration process.

The integration layer defines an external call from the pre transformation extension point. This extension point accepts source XML as input and gives the source XML as output. The implementation can choose to plug in a concrete WSDL instead of the abstract WSDL. This can assist the implementation in invoking any external Web service and transform the input XML.

5.1.2 Post transformation Extension Point

The post transformation extension point is invoked after the main transformation is executed. This transformation aids in transforming the target XML going as an input to the target queue.

The integration layer defines an external call from the post transformation extension point. This extension point accepts the target XML as input and gives the target XML as output. The implementation can choose to plug in a concrete WSDL instead of the abstract WSDL. This can assist the implementation in invoking any external Web service and transform the output XML.

5.1.3 Custom Transformations

The custom transformations are used to add data to custom elements in the incoming and outgoing messages. The incoming and outgoing messages have custom elements defined in the message. These custom elements refer to a custom XML schema. The main transformation invokes custom transformation.

Empty custom transformation and custom schemas are shipped with the product. The implementation team can add additional fields in the custom schema and map them using the custom transformations.

Using custom transformations allows the implementation to define and pass additional data from the source system to the target system.

5.1.4 Steps to Implement Extension Points

To implement extension points, perform the following:

1. Each process in the integration has a pre- and post-transformation extension point which can be used to invoke Web services and transform the payload.
2. The desired extension point can be triggered from the process by enabling it using the ConfigurationProperties.xml pre- and post-transformation extension flags as described in section Setting Configuration Properties.
3. Each process has its own concrete wsdl which is used to read the endpoint location for the extension service.
   These concrete wsdl files are located in MDS under the following directories:
   - /apps/ODM-MDM/AIAMetaData/AIAComponents/ExtensionServiceLibrary/OUODM
   - /apps/ODM-MDM/AIAMetaData/AIAComponents/ExtensionServiceLibrary/OUMDM

   Update the concrete wsdl file to soap address location details for the extension service to be called and move the concrete wsdl file to MDS.
4. To move the concrete wsdl to MDS, update the appropriate wsdl in the product install home. The directories to put the concrete wsdl in product install home are the following:
   - $PRODUCT_HOME/MDS-Artifacts/ODM-MDM/AIAMetaData/AIAComponents/ExtensionServiceLibrary/OUODM
   - $PRODUCT_HOME/MDS-Artifacts/ODM-MDM/AIAMetaData/AIAComponents/ExtensionServiceLibrary/OUMDM

   Then deploy the concrete wsdls to MDS by running the ant deploy command for Deploying MDS folder.

For more information regarding the command for deploying MDS folder, refer to the Oracle Utilities Integration for Device Operations Installation Guide “Deploying MDS Folder” section.

5. After deploying the files to MDS, restart the SOA server.
6. After restarting the SOA server, the extension point invokes the Web service in the concrete WSDL.
   For example: To enable the extension points for ODMMDMAssetDeviceSyncReqExtension, change the soap address ' location and set the URL of the extension service.

```
<service name="ODMMDMAssetDeviceSyncReqExtensionAbstractService">
    <port name="ODMMDMAssetDeviceSyncReqExtensionAbstractServicePort"

binding="tns:ODMMDMAssetDeviceSyncReqExtensionAbstractServiceBinding">
      <soap:address location="<URL of the extension service>"/>
    </port>
  </service>
```

> **Note**: The binding and service can be added easily using the Oracle Jdeveloper 11g.

5.1.5 Steps to Implement Custom Transformations

To implement custom transformations, perform the following:

1. Each process in the integration has its own XSD file. The messages have custom elements which can be used to pass additional data. Refer to message mappings to see the location of customElements in each message.
2. Each XSD has a corresponding CustomType xsd file in which the complexType elements for each customElements tag are defined.
3. Each process uses two XSD files, one for the Oracle Utilities Operational Device Management message and one for the Oracle Utilities Meter Data Management message.
4. To pass additional elements in the customElements tag, the corresponding complexType needs to be modified. Add the additional elements required in both complexType elements, that is, xsd for both edge applications.
5. Each process has a main transformation which invokes custom templates. Each main transformation file has a corresponding custom XSL and custom templates are defined in the custom XSL.
6. These custom templates are invoked at the location where each customElements tag is present.
7. The custom XSL can be modified to add transformation for the newly added elements in custom XSD files.
8. The custom XSD files are located in product install home under the following directories:
   - ODMMDM/MDS-Artifacts/ODM-MDM/AIAMetaData/AIAComponents/ApplicationObjectLibrary/OUODM/V2/schemas
   - ODMMDM/MDS-Artifacts/ ODM-MDM/AIAMetaData/AIAComponents/ApplicationObjectLibrary/OUMDM/V2/schemas

   The custom XSL files are located in product install home under the directory
   - ODM-MDM/services/industry/Utilities/EBF/<Process Name>/xsl

9. After updating the XSD and XSL files in the product install home, update MDS using the ant deploy command for Deploying MDS folder.

For more information about the command to for deploying MDS, refer to the Oracle Utilities Integration for Device Operations Installation Guide "Deploying MDS Folder" section.

10. After deploying the files to MDS, restart the SOA server.

11. After restarting the SOA server, the changes to the custom xsd and xsl will be reflected in the integration.

    For example: In the Asset-Device, synchronization process to pass **sendDetails > finalSnapshot > customElements > distMethod element in Oracle Utilities Operational Device Management to SyncRequestDetails > finalSnapshot > customElements > distributionMethod** element in MDM, the following changes need to be done:

    - "In ODMAssetSyncRequestCustom.xsd, add the distMethod element to the schema. This xsd file is located in ODM-MDM/MDS-Artifacts/ODM-MDM/AIAMetaData/AIAComponents/ ApplicationObjectLibrary/OUODM/V2/schemas folder.

```
<xsd:complexType name="finalSnapshotCustomType">
  <xsd:sequence>
     <xsd:element name="distMethod" type="xsd:string"/>
  </xsd:sequence>
<xsd:complexType>
```

    - In MDMDeviceSyncRequestCustom.xsd, add the distributionMethod element in the schema. This xsd file is located in ODM-MDM/MDS-Artifacts/ODM-MDM/AIAMetaData/AIAComponents/ ApplicationObjectLibrary/OUMDM/V2/schemas folder.

```
<xsd:complexType name="finalSnapshotCustomType">
  <xsd:sequence>
     <xsd:element name="distributionMethod" type="xsd:string"/>
  </xsd:sequence>
<xsd:complexType>
```

    - "Transformation Xform_ODMAssetSyncReq_To_MDMDeviceSyncReq_Custom.xsl

```
 <xsl:template name="finalSnapshot-customElements">
 <!-- This template is use for Xformation of
    //finalSnapshot/customElements in Request Message-->
  <distributionMethod xmlns="http://ugbu.oracle.com/MDMRequest">
    <xsl:value-of   select="/ns0:sendDetails/ns0:syncRequestDetails/
ns0:finalSnapshot/ns0:customElements/ns0:distMethod"/>
  </distributionMethod>
</xsl:template>
```

Appendix A

Data Mapping

This section provides mapping details for each integration point. They are as follows:

- Device Synchronization
- SP-Location Synchronization
- Contact Synchronization
- Install Event - Asset Location Synchronization

Device Synchronization

This section discusses in detail the following:

- Device Sync Request Mapping
- Device Sync Response Mapping

Device Sync Request Mapping

Device Sync Request Mapping details for each integration point are as shown in the table below:

ODM Asset Request Message			MDM Device Request Message			DVM Mapping		
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**	**ODM Column**	**MDM Column**
sendDetails		Outermost Tag	D1-SyncRequestInbound		OutermostTag			
syncRequestId	sendDetails	Field	externalReferenceId	D1-SyncRequestInbound	Field			
			syncRequestId	D1-SyncRequestInbound	Field			
			bo	D1-SyncRequestInbound	Field			

(Continued)ODM Asset Request Message			MDM Device Request Message			DVM Mapping		
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**	**ODM Column**	**MDM Column**
			boStatus	D1-SyncRequestInbound	Field			
			createDateTime	D1-SyncRequestInbound	Field			
			statusDateTime	D1-SyncRequestInbound	Field			
initialLoad	sendDetails	Field	initialLoad	D1-SyncRequestInbound	Field			
mo	sendDetails	Field	targetMo	D1-SyncRequestInbound	Field	ODM_MDM_MO	ODM_MO	MDM_MO
sourceSystem	sendDetails	Field	externalSystem	D1-SyncRequestInbound	Field			
pkValue1	sendDetails	Field	externalPkValue1	D1-SyncRequestInbound	Field			
pkValue2	sendDetails	Field	externalPkValue2	D1-SyncRequestInbound	Field			
pkValue3	sendDetails	Field	externalPkValue3	D1-SyncRequestInbound	Field			
pkValue4	sendDetails	Field	externalPkValue4	D1-SyncRequestInbound	Field			
pkValue5	sendDetails	Field	externalPkValue5	D1-SyncRequestInbound	Field			
			productionPkValue	D1-SyncRequestInbound	Field			
			version	D1-SyncRequestInbound	Field			
			relatedCompositeSyncId	D1-SyncRequestInbound	Field			
			clearExceptions	D1-SyncRequestInbound	Field			

(Continued)ODM Asset Request Message			MDM Device Request Message			DVM Mapping		
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**	**ODM Column**	**MDM Column**
			targetBo	D1-SyncRequestInbound	Field			
bo	sendDetails	Field						
boStatus	sendDetails	Field						
createDateTime	sendDetails	Field						
statusDateTime	sendDetails	Field						
version	sendDetails	Field						
syncRequired	sendDetails	Field						
forceSync	sendDetails	Field						
initialLoad	sendDetails	Field						
discardReason	sendDetails	Field						
cancelReason	sendDetails	Field						
syncRequestDetails	sendDetails	Group	syncRequestDetails	D1-SyncRequestInbound	Group			
			original	syncRequest Details	Group			
initialSnapshot	syncRequestDetails	Group	initialSnapshot	original	Group			
deviceType	initialSnapshot	Field	deviceType	initialSnapshot	Field	MDM_ODM_AssetDeviceType	ODM_AssetType	MDM_DeviceType
boStatus	initialSnapshot	Field	boStatus	initialSnapshot	Field	MDM_ODM_AssetDeviceStatus	ODM_AssetStatus	MDM_DeviceStatus
			manufacturer	initialSnapshot	Field			
			model	initialSnapshot	Field			
customElements	initialSnapshot	Field	customElements	initialSnapshot	Field			
formattedElements	initialSnapshot	Field	formattedElements	initialSnapshot	Field			
finalSnapshot	syncRequestDetails	Group	finalSnapshot	original	Group			

(Continued)ODM Asset Request Message			MDM Device Request Message			DVM Mapping		
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**	**ODM Column**	**MDM Column**
deviceType	finalSnapshot	Field	deviceType	finalSnapshot	Field	MDM_ODM_AssetDeviceType	ODM_AssetType	MDM_DeviceType
boStatus	finalSnapshot	Field	boStatus	finalSnapshot	Field	MDM_ODM_AssetDeviceStatus	ODM_AssetStatus	MDM_DeviceStatus
			manufacturer	finalSnapshot	Field			
			model	finalSnapshot	Field			
customElements	finalSnapshot	Field	customElements	finalSnapshot	Field			
formattedElements	finalSnapshot	Field	formattedElements	finalSnapshot	Field			

Device Sync Response Mapping

Device Sync Response Mapping details for each integration point are as shown in the table below:

ODM Asset Request Message			MDM Device Request Message			DVM Mapping		
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**	**MDM Column**	**ODM Column**
sendDetails		OutermostTag	F1-UpdateAndTransitionSyncRequest		Outermost Tag			
externalReferenceId	sendDetails	Field	syncRequestId	F1-UpdateAndTransitionSyncRequest	Field			
externalIds	sendDetails	Group	externalIds	F1-UpdateAndTransitionSyncRequest	Group			
entityIdList	externalIds	List	entityIdList	externalIds	List			
entity	entityIdList	Field	entity	entityIdList	Field			
externalPkValue1	entityIdList	Field	externalPk1	entityIdList	Field			
externalPkValue2	entityIdList	Field	externalPk2	entityIdList	Field			
externalPkValue3	entityIdList	Field	externalPk3	entityIdList	Field			
externalPkValue4	entityIdList	Field	externalPk4	entityIdList	Field			
externalPkValue5	entityIdList	Field	externalPk5	entityIdList	Field			
exceptionInformation	sendDetails	Group						

(Continued)ODM Asset Request Message			MDM Device Request Message			DVM Mapping		
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**	**MDM Column**	**ODM Column**
exceptionInformationList	exceptionInformation	List	exceptionInfo	F1-UpdateAndTransitionSyncRequest	Group			
messageCategory	exceptionInformationList	Field	messageCategory	exceptionInfo	Field			
messageNumber	exceptionInformationList	Field	messageNumber	exceptionInfo	Field	MDM_ODM_ErrorCode	MDM_ErrorCode	ODM_ErorCode
sequence	exceptionInformationList	Field	sequence	exceptionInfo	Field			
comments	exceptionInformationList	Field	comments	exceptionInfo	Field			
messageParameters	exceptionInformationList	List	messageParameters	exceptionInfo	List			
parameterSequence	messageParameters	Field	parameterSequence	messageParameters	Field			
messageParameterValue	messageParameters	Field	messageParameterValue	messageParameters	Field			
messageParameterType	messageParameters	Field						
customElements	sendDetails	Field	customElements	F1-UpdateAndTransitionSyncRequest	Field			

SP-Location Synchronization

SP- Location Synchronization for each integration point consists of the following:

- SP-Location Sync Request Mapping
- SP-Location Sync Response Mapping

SP-Location Sync Request Mapping

SP Location Sync Request Mapping for each integration point is shown in the table below:

MDM SP Request Message			ODM Location Request Message			DVM Mapping		
Element Name	Parent Element	Type	Element Name	Parent Element	Type	DVM	ODM Column	MDM Column
sendDetail		OutermostTag	W1-SyncRequestInbound		OutermostTag			
syncRequestId	sendDetail	Field	externalReferenceId	W1-SyncRequestInbound	Field			
			syncRequestId	W1-SyncRequestInbound	Field			
			bo	W1-SyncRequestInbound	Field			
			boStatus	W1-SyncRequestInbound	Field			
			createDateTime	W1-SyncRequestInbound	Field			
			statusDateTime	W1-SyncRequestInbound	Field			
initialLoad	sendDetail	Field	initialLoad	W1-SyncRequestInbound	Field			
mo	sendDetail	Field	targetMo	W1-SyncRequestInbound	Field	MDM_ODM_MO	ODM_MO	MDM_MO
sourceSystem	sendDetail	Field	externalSystem	W1-SyncRequestInbound	Field			
pkValue1	sendDetail	Field	externalPkValue1	W1-SyncRequestInbound	Field			
pkValue2	sendDetail	Field	externalPkValue2	W1-SyncRequestInbound	Field			

(Continued)MDM SP Request Message			ODM Location Request Message			DVM Mapping		
Element Name	Parent Element	Type	Element Name	Parent Element	Type	DVM	ODM Column	MDM Column
pkValue3	sendDetail	Field	externalPkValue3	W1-SyncRequestInbound	Field			
pkValue4	sendDetail	Field	externalPkValue4	W1-SyncRequestInbound	Field			
pkValue5	sendDetail	Field	externalPkValue5	W1-SyncRequestInbound	Field			
			productionPkValue	W1-SyncRequestInbound	Field			
			version	W1-SyncRequestInbound	Field			
			relatedCompositeSyncId	W1-SyncRequestInbound	Field			
			clearExceptions	W1-SyncRequestInbound	Field			
			targetBo	W1-SyncRequestInbound	Field			
bo	sendDetail	Field						
boStatus	sendDetail	Field						
createDateTime	sendDetail	Field						
statusDateTime	sendDetail	Field						
version	sendDetail	Field						
syncRequired	sendDetail	Field						
forceSync	sendDetail	Field						
initialLoad	sendDetail	Field						
discardReason	sendDetail	Field						
cancelReason	sendDetail	Field						
externalIds	sendDetail	Group						
externalIdsList	externalIds	List						
entity	externalIdsList	Field						
externalPkValue1	externalIdsList	Field						
externalPkValue2	externalIdsList	Field						

(Continued)MDM SP Request Message			ODM Location Request Message			DVM Mapping		
Element Name	Parent Element	Type	Element Name	Parent Element	Type	DVM	ODM Column	MDM Column
externalPkValue3	externalIdsList	Field						
externalPkValue4	externalIdsList	Field						
externalPkValue5	externalIdsList	Field						
syncRequestDetails	sendDetail	Group	syncRequestDetails	W1-SyncRequestInbound	Group			
			original	syncRequestDetails	Group			
initialSnapshot	syncRequestDetails	Group	initialSnapshot	syncRequestDetails	Group			
nodeType	initialSnapshot	Field	nodeTypei	initialSnapshot	Field	MDM_ODM_SPNodeType	ODM_NodeType	MDM_SPType
nodeDisposition	initialSnapshot	Field	nodeDisposition	initialSnapshot	Field	MDM_ODM_NodeDisposition	ODM_NodeDisposition	MDM_NodeDisposition
disconnectLocation	initialSnapshot	Field	disconnectLocation	initialSnapshot	Field	MDM_ODM_DisconnectLocation	ODM_DisconnectLocation	MDM_DisconnectLocation
okToEnter	initialSnapshot	Field	okToEnter	initialSnapshot	Field	MDM_ODM_OkToEnterCode	ODM_OkToEnterCode	MDM_OkToEnterCode
location	initialSnapshot	Group	location	initialSnapshot	Group			
country	location	Field	country	location	Field	MDM_ODM_Country	ODM_CountryCode	MDM_CountryCode
lifeSupportSensitiveLoad	initialSnapshot	Field	lifeSupportSensitiveLoad	initialSnapshot	Field	MDM_ODM_LifeSupportSensitiveLoad	ODM_LifeSupportSensitiveLoad	MDM_LifeSupportSensitiveLoad
timeZone	initialSnapshot	Field	timeZone	initialSnapshot	Field	MDM_ODM_TimeZone	ODM_TimeZone	MDM_TimeZone

(Continued)MDM SP Request Message			ODM Location Request Message			DVM Mapping		
Element Name	Parent Element	Type	Element Name	Parent Element	Type	DVM	ODM Column	MDM Column
customElements	initialSnapshot	Field	customElements	initialSnapshot	Field			
formattedElements	initialSnapshot	Field	formattedElements	initialSnapshot	Field			
finalSnapshot	syncRequestDetails	Group	finalSnapshot	syncRequestDetails	Group			
nodeType	initialSnapshot	Field	nodeType	initialSnapshot	Field	MDM_ODM_SPNodeType	ODM_NodeType	MDM_SPType
nodeDisposition	initialSnapshot	Field	nodeDisposition	initialSnapshot	Field	MDM_ODM_NodeDisposition	ODM_NodeDisposition	MDM_NodeDisposition
disconnectLocation	initialSnapshot	Field	disconnectLocation	initialSnapshot	Field	MDM_ODM_DisconnectLocation	ODM_DisconnectLocation	MDM_DisconnectLocation
okToEnter	initialSnapshot	Field	okToEnter	initialSnapshot	Field	MDM_ODM_OkToEnterCode	ODM_OkToEnterCode	MDM_OkToEnterCode
location	initialSnapshot	Group	location	initialSnapshot	Group			
country	location	Field	country	location	Field	MDM_ODM_Country	ODM_CountryCode	MDM_CountryCode
lifeSupportSensitiveLoad	initialSnapshot	Field	lifeSupportSensitiveLoad	initialSnapshot	Field	MDM_ODM_LifeSupportSensitiveLoad	ODM_LifeSupportSensitiveLoad	MDM_LifeSupportSensitiveLoad
timeZone	initialSnapshot	Field	timeZone	initialSnapshot	Field	MDM_ODM_TimeZone	ODM_TimeZone	MDM_TimeZone
customElements	initialSnapshot	Field	customElements	initialSnapshot	Field			
formattedElements	initialSnapshot	Field	formattedElements	initialSnapshot	Field			

SP-Location Sync Response Mapping

SP-Location Sync Response Mapping details for each integration point are as shown in the table below:

ODMLocation Sync Response Message			MDM SP Sync Response Message			DVM Mapping		
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**	**MDM Column**	**ODM Column**
sendDetail		OutermostTag	F1-UpdateAndTransitionSyncRequest		OutermostTag			
externalReferenceId	sendDetails	Field	syncRequestId	F1-UpdateAndTransitionSyncRequest	Field			
externalIds	sendDetails	Group	externalIds	F1-UpdateAndTransitionSyncRequest	Group			
entityIdList	externalIds	List	entityIdList	externalIds	List			
entity	entityIdList	Field	entity	entityIdList	Field			
externalPk1	entityIdList	Field	externalPk1	entityIdList	Field			
externalPk2	entityIdList	Field	externalPk2	entityIdList	Field			
externalPk3	entityIdList	Field	externalPk3	entityIdList	Field			
externalPk4	entityIdList	Field	externalPk4	entityIdList	Field			
externalPk5	entityIdList	Field	externalPk5	entityIdList	Field			
exceptionInformation	sendDetails	Group						
exceptionInformationList	exceptionInformation	List	exceptionInfo	F1-UpdateAndTransitionSyncRequest	List			
messageCategory	exceptionInformationList	Field	messageCategory	exceptionInfo	Field	MDM_ODM_ErrorCode	MDM_ErrorCode	ODM_ErrorCode
messageNumber	exceptionInformationList	Field	messageNumber	exceptionInfo	Field	MDM_ODM_ErrorCode	MDM_ErrorCode	ODM_ErrorCode
sequence	exceptionInformationList	Field	sequence	exceptionInfo	Field			
comments	exceptionInformationList	Field	comments	exceptionInfo	Field			
messageParameters	exceptionInformationList	List	messageParameters	exceptionInfo	List			

(Continued)ODMLocation Sync Response Message			MDM SP Sync Response Message			DVM Mapping		
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**	**MDM Column**	**ODM Column**
parameterSequence	messageParameters	Field	parameterSequence	messageParameters	Field			
messageParameterValue	messageParameters	Field	messageParameterValue	messageParameters	Field			
messageParameterType	messageParameters	Field						
customElements	customElements sendDetails	Field	customElements	F1-UpdateAndTransitionSyncRequest	Field			

Contact Synchronization

Contact Synchronization sync request and response mapping includes the following topics:

- Contact SyncRequestMapping
- ContactSyncResponseMapping

Contact SyncRequestMapping

Details for Contact Synchronization are as shown in the table below:

MDMContactRequestMessage			ODMContactRequest Message			DVM Mapping		
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**	**ODM**	**MDM**
sendDetails		OutermostTag	W1-SyncRequestInbound		OutermostTag			
syncRequestId	sendDetails	Field	externalReferenceId	W1-SyncRequestInbound	Field			
			syncRequestId	W1-SyncRequestInbound	Field			
			bo	W1-SyncRequestInbound	Field			
			boStatus	W1-SyncRequestInbound	Field			
			createDateTime	W1-SyncRequestInbound	Field			
			statusDateTime	W1-SyncRequestInbound	Field			
initialLoad	sendDetails	Field	initialLoad	W1-SyncRequestInbound	Field			
mo	sendDetails	Field	targetMo	W1-SyncRequestInbound	Field	MDM_ODM_MO	ODM_MO	MDM_MO
sourceSystem	sendDetails	Field	externalSystem	W1-SyncRequestInbound	Field			
pkValue1	sendDetails	Field	externalPkValue1	W1-SyncRequestInbound	Field			
pkValue2	sendDetails	Field	externalPkValue2	W1-SyncRequestInbound	Field			

(Continued)MDMContactRequestMessage			ODMContactRequest Message			DVM Mapping		
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**	**ODM**	**MDM**
pkValue3	sendDetails	Field	externalPkValue3	W1-SyncRequestInbound	Field			
pkValue4	sendDetails	Field	externalPkValue4	W1-SyncRequestInbound	Field			
pkValue5	sendDetails	Field	externalPkValue4	W1-SyncRequestInbound	Field			
			productionPkValue	W1-SyncRequestInbound	Field			
			version	W1-SyncRequestInbound	Field			
			relatedCompositeSyncId	W1-SyncRequestInbound	Field			
			clearExceptions	W1-SyncRequestInbound	Field			
			targetBo	W1-SyncRequestInbound	Field			
bo	sendDetails	Field						
boStatus	sendDetails	Field						
createDateTime	sendDetails	Field						
statusDateTime	sendDetails	Field						
Version	sendDetails	Field						
syncRequired	sendDetails	Field						
forceSync	sendDetails	Field						
initialLoad	sendDetails	Field						
discardReason	sendDetails	Field						
cancelReason	sendDetails	Field						
mo	sendDetails	Field						
syncRequestDetails	sendDetails	Group	syncRequestDetails	W1-SyncRequestInbound	Group			
			original	syncRequestDetails	Group			
initialSnapshot	original	Group	initialSnapshot	original	Group			

(Continued)MDMContactRequestMessage			ODMContactRequest Message			DVM Mapping		
contactType	initialSnapshot	Field	contactType	nitialSnapshot	Field	MDM_ ODM_ Contact Type	ODM_ Contact Type	MDM_ Contact Type
contactType	initialSnapshot	Field	contactType	nitialSnapshot	Field	MDM_ ODM_ Contact Type	ODM_ Contact Type	MDM_ Contact Type
customElements	initialSnapshot	Field	customElements	initialSnapshot	Field			
formattedElements	initialSnapshot	Field	formattedElements	initialSnapshot	Field			
finalSnapshot	original	Group	finalSnapshot	syncRequestDetails	Group			
contactType	finalSnapshot	Field	contactType	finalSnapshot	Field	MDM_ ODM_ Contact Type	ODM_ Contact Type	MDM_ Contact Type
customElements	finalSnapshot	Field	customElements	finalSnapshot	Field			
formattedElements	finalSnapshot	Field	formattedElements	finalSnapshot	Field			

ContactSyncResponseMapping

Details for Contact Synchronization integration point are as shown in the table below:

ODMContactResponseMessage			MDMContactResponse Message			DVM Mapping		
Element Name	Parent Element	Type	Element Name	Parent Element	Type	DVM	ODM	MDM
sendDetails		Outermost Tag	F1-UpdateAndTransitionSyncRequest		Outermost Tag			
externalReferenceId	sendDetails	Field	syncRequestId	F1-UpdateAndTransitionSyncRequest	Field			
externalIds	sendDetails	Group	externalIds	F1-UpdateAndTransitionSyncRequest	Group			
entityIdList	externalIds	List	entityIdList	externalIds	List			
entity	entityIdList	Field	entity	entityIdList	Field			
externalPk1	entityIdList	Field	externalPk1	entityIdList	Field			
externalPk2	entityIdList	Field	externalPk2	entityIdList	Field			
externalPk3	entityIdList	Field	externalPk3	entityIdList	Field			

			MDMContactResponse Message			DVM Mapping		
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**	**ODM**	**MDM**
externalPk4	entityIdList	Field	externalPk4	entityIdList	Field			
externalPk5	entityIdList	Field	externalPk5	entityIdList	Field			
exceptionInformation	sendDetails	Group						
exceptionInformationList	exceptionInformation	List	exceptionInfo	F1-UpdateAndTransitionSyncRequest	List			
messageCategory	exceptionInformationList	Field	messageCategory	exceptioninfo	Field			
messageNumber	exceptionInformationList	Field	messageNumber	exceptioninfo	Field	MDM_ODM_Error Code	MDM_Error Code	ODM_Error Code
sequence	exceptionInformationList	Field	sequence	exceptioninfo	Field			
comments	exceptionInformationList	Field	comments	exceptioninfo	Field			
messageParameters	exceptionInformationList	List	messageParamaters	exceptioninfo	List			
parameterSequence	messageParameters	Field	parameterSequence	messageParameters	Field			
messageParameterValue	messageParameters	Field	messageParameterValue	messageParameters	Field			
messageParameterType	messageParameters	Field						
customElements	sendDetails	Field	customElements	F1-UpdateAndTransitionSyncRequests	Field			

Install Event - Asset Location Synchronization

Install Event- Asset Node Synchronization has mapping details for each integration point for the following topics:

- Install Event - Asset Location Sync Request Mapping
- Install Event - Asset Location Sync Response Mapping

Install Event - Asset Location Sync Request Mapping

Details for Install Event - Asset Location are as shown in the table below:

MDMInstallEventRequestMessage			ODMAssetLocationRequestMessages			DVM Mapping		
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**	**ODM**	**MDM**
sendDetails		OutermostTag	W1-SyncRequestInboundAssetNode		OutermostTag			
syncRequestId	sendDetails	Field	externalReferenceId	W1-SyncRequestInboundAssetNode	Field			
			syncRequestId	W1-SyncRequestInboundAssetNode	Field			
			bo	W1-SyncRequestInboundAssetNode	Field			
			boStatus	W1-SyncRequestInboundAssetNode	Field			
			createDateTime	W1-SyncRequestInboundAssetNode	Field			
			statusDateTime	W1-SyncRequestInboundAssetNode	Field			
initialLoad	sendDetails	Field	initialLoad	W1-SyncRequestInboundAssetNode	Field			
mo	sendDetails	Field	targetMo	W1-SyncRequestInboundAssetNode	Field			

			ODMAssetLocationRequestMessages			DVM Mapping		
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**	**ODM**	**MDM**
sourceSystem	sendDetails	Field	externalSystem	W1-SyncRequestInboundAssetNode	Field			
pkValue1	sendDetails	Field	externalPkValue1	W1-SyncRequestInboundAssetNode	Field	MDM_ODM_MO	ODM_MO	MDM_MO
pkValue2	sendDetails	Field	externalPkValue2	W1-SyncRequestInboundAssetNode	Field			
pkValue3	sendDetails	Field	externalPkValue3	W1-SyncRequestInboundAssetNode	Field			
pkValue4	sendDetails	Field	externalPkValue4	W1-SyncRequestInboundAssetNode	Field			
pkValue5	sendDetails	Field	externalPkValue5	W1-SyncRequestInboundAssetNode	Field			
			productionPkValue	W1-SyncRequestInboundAssetNode	Field			
			vesion	W1-SyncRequestInboundAssetNode	Field			
			relatedCompositeSyncId	W1-SyncRequestInboundAssetNode	Field			
			clearExceptions	W1-SyncRequestInboundAssetNode	Field			
			targetBo	W1-SyncRequestInboundAssetNode	Field			
bo	sendDetail	Field						
boStatus	sendDetail	Field						
createDateTime	sendDetail	Field						
statusDateTime	sendDetail	Field						

			ODMAssetLocationRequestMessages			DVM Mapping		
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**	**ODM**	**MDM**
version	sendDetail	Field						
syncRequired	sendDetail	Field						
forceSync	sendDetail	Field						
initialLoad	sendDetail	Field						
discardReason	sendDetail	Field						
cancelReason	sendDetail	Field						
mo	sendDetail	Field						
syncRequestDetails	sendDetail	Group	syncRequestDetails	W1-SyncRequestInboundAssetNode	Group			
			original	syncRequestDetails	Group			
initialSnapshot	original	Group	initialSnapshot	syncRequestDetails	Group			
customElements	initialSnapshot	Field	customElements	initialSnapshot	Field			
formattedElements	initialSnapshot	Field	formattedElements	initialSnapshot	Field			
finalSnapshot	original	Group	finalSnapshot	syncRequestDetails	Group			
customElements	finalSnapshot	Field	customElements	finalSnapshot	Field			
formattedElements	finalSnapshot	Field	formattedElements	finalSnapshot	Field			
createToDo	syncRequestDetails	Field						
assetMovement	syncRequestDetails	Group	assetMovement	syncRequestDetails	Group			

Install Event - Asset Location Sync Response Mapping

Install Event Asset Location Sync Response mapping details are shown in the table below:

ODM Asset Node Response Message			MDM Install Event Response Message			DVM Mapping		
Element Name	Parent Element	Type	Element Name	Parent Element	Type	DVM	ODM	MDM
sendDetails		Outermost Tag	F1-UpdateAndTransitionSyncRequest		OutermostTag			
externalReferenceId	sendDetails	Field	syncRequestId	F1-UpdateAndTransitionSyncRequest	Field			
externalIds	sendDetails	Group	externalIds	F1-UpdateAndTransitionSyncRequest	Group			
entityIdList	externalIds	List	entityIdList	externalIds	List			
entity	entityIdList	Field	entity	entityIdList	Field			
externalPk1	entityIdList	Field	externalPk1	entityIdList	Field			
externalPk2	entityIdList	Field	externalPk2	entityIdList	Field			
externalPk3	entityIdList	Field	externalPk3	entityIdList	Field			
externalPk4	entityIdList	Field	externalPk4	entityIdList	Field			
externalPk5	entityIdList	Field	externalPk5	entityIdList	Field			
exceptionInformation	sendDetails	Group						
exceptionInformationList	exceptionInformation	List	exceptionInfo	F1-UpdateAndTransitionSyncRequest	List			
messageCategory	exceptionInformationList	Field	messageCategory	exceptionInfo	Field			
messageNumber	exceptionInformationList	Field	messageNumber	exceptionInfo	Field	MDM_ODM_ErrorCode	MDM_ErrorCode	ODM_ErorCode
sequence	exceptionInformationList	Field	sequence	exceptionInfo	Field			
comments	exceptionInformationList	Field	comments	exceptionInf	Field			
messageParameters	exceptionInformationList	List	messageParameters	exceptioninfo	List			
parameterSequence	messageParameters	Field	parameterSequence	messageParameters	Field			

ODM Asset Node Response Message			MDM Install Event Response Message			DVM Mapping		
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**	**ODM**	**MDM**
messageParameterValue	messageParameters	Field	messageParameterValue	messageParameters	Field			
messageParameterType	messageParameters	Field						
customElements	sendDetails	Field	customElements	F1-UpdateAndTransitionSyncRequest	Field			

Appendix B

Cross References

The following sections provide references for where you can find more information on some of the terms and entities related to this integration.

- Domain Value Maps
- JMS Adapter

Domain Value Maps

Refer to the chapters titled *Working with Domain Value Maps* and *Using SOA Composer with Domain Value Maps* in Oracle® Fusion Middleware Developer's Guide for Oracle SOA Suite.

JMS Adapter

Refer to the *Oracle Fusion Middleware User's Guide* for Technology Adapters for more information.

Index