Oracle® Application Integration Architecture

Pre-Built Integrations: Functional Interoperability Configuration Guide

Release 11.3

E41648-01

December 2013

Documentation for administrators that describes the functional interoperability of Application Integration Architecture (AIA) pre-built integrations. Specifically, it includes configuration steps needed for functional interoperability for Agile Product Lifecycle Management Integration Pack for Oracle E-Business Suite: Design to Release, Siebel CRM Integration Pack for Oracle Order Management: Order to Cash, Oracle Customer Master Data Management (MDM) Integration Base Pack, Oracle Customer MDM Integration Option for Siebel CRM, Oracle Customer MDM Integration Option for Oracle E-Business Suite, Oracle Customer MDM Integration Option for Oracle Communications BRM, Oracle Product MDM Integration Base Pack, Oracle Product MDM Integration Option for Siebel CRM, Oracle Product MDM Integration Option for Oracle Communications BRM, Oracle Communications Order to Cash - Siebel CRM Option and BRM Option, Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care and Oracle Communications Order to Cash Integration Pack for Siebel CRM, Oracle OSM, and Oracle BRM.

ORACLE®

Oracle Application Integration Architecture Pre-Built Integrations: Functional Interoperability Configuration Guide, Release 11.3

E41648-01

Contents

5 Product MDM with Order to Cash

6 Customer MDM with Communications Order to Cash

7 Customer MDM with Communications Agent Assisted Billing Care

8 Product MDM with Communications Order to Cash

List of Examples

List of Figures

List of Tables

Preface

Welcome to the Pre-Built Integrations: Functional Interoperability Configuration Guide.

What's New in this Guide

■ The Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations is restructured into a general installation chapter with an individual configuration and deployment chapter for each pre-built integration.

■ The term *process integration pack* is replaced with the term *pre-built integrations*.

■ The implementation guides are restructured into two parts: design and set up.

Part I - Design: This part provides functional overviews, activity diagrams, assumptions and constraints, and technical sequence diagrams and steps.

Part II - Set up: This part provides prerequisites, data requirements, and configuration steps.

■ Starting with this release, these integrations are no longer available:

– Oracle CRM On Demand Integration Pack for JD Edwards EnterpriseOne: Lead to Order

– Oracle Workforce Administration Integration Pack for PeopleSoft Human Resources

Common Oracle AIA Pre-Built Integration Guides

Oracle Application Integration Architecture Pre-Built Integrations 11.3 includes the following guides shared by all products delivered with this release:

■ Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations Release 11.5

This guide provides an overview of the installation process, including how to install, configure, and deploy your pre-built integrations. The steps required to upgrade your pre-built integrations to the latest release are also provided.

■ Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide

This guide describes:

– How to work with and configure Session Pool Manager (SPM), which is a service in the Oracle SOA Suite web server whose primary function is to manage a pool of web server session tokens that can be reused by BPEL flows.

- How to deploy and configure the AIACompositeScheduler. This is a utility component that is used by pre-built integrations to schedule a service-oriented architecture (SOA) composite to be invoked at the specified time interval.

- Oracle Application Integration Architecture Pre-Built Integrations 11.5: Product-to-Guide Index

  The Product-to-Guide index lists the guides that provide information for each product delivered in this release.

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

Master Notes

Master notes provide a one-stop reference point for the following:

- Latest manuals

- Alerts

- Troubleshooting details

- FAQs

- Patching information

- AIA community link and more.

For master notes for the pre-built integrations, visit Master Note ID 1318600.1 on My Oracle Support (MOS): http://support.oracle.com.

Additional Resources

The following resources are also available:

- **Oracle Application Integration Architecture Foundation Pack:**

  Oracle AIA Pre-Built integrations require Foundation Pack 11.1.1.5.0 to be installed. Refer to the Foundation Pack documentation library on OTN to download the Foundation Pack guides at http://download.oracle.com/docs/cd/E21764_01/aia.htm.

- **Oracle Application Integration Architecture: Product-to-Guide Index:**

  Oracle Technology Network: http://www.oracle.com/technetwork/index.html

- **Known Issues and Workarounds:**

  My Oracle Support: https://support.oracle.com/

- **Release Notes:**

  Oracle Technology Network:
  <http://www.oracle.com/technetwork/index.html>

- **Documentation updates:**

  Oracle Technology Network:
  <http://www.oracle.com/technetwork/index.html>

1

Understanding Functional Interoperability

This guide describes the functional interoperability of Oracle Application Integration Architecture (Oracle AIA) pre-built integrations. This chapter includes a functional interoperability matrix that lists the interoperable integrations included in the Oracle AIA 11.5 release.

This chapter includes the following sections:

1.1 Overview

A single Oracle AIA release contains multiple pre-built integrations, enabling customers to deploy multiple integrations on a single Oracle AIA instance (cluster) if desired. Functional interoperability is defined as the ability to deploy two or more integrations on the same Oracle AIA instance (cluster) and for the services provided by the integrations to work together to fulfill end-to-end business processes. Functionally interoperable integrations have also been tested together by Oracle AIA Development.

Depending on the deployment size, mainly driven by throughput and availability requirements, Oracle recommends that tier 1 and 2 customers split up the different integrations in separate Oracle AIA instances (clusters). This document refers specifically to deployments where multiple pre-built integrations are deployed on a single Oracle AIA instance, therefore requiring functional interoperability.

Each of the integrations has been designed to support a specific set of source and target applications. As delivered, the routing rules for each of these integrations point to specific target applications. However, when a customer deploys multiple integrations, these routing rules must be modified so that the integrations can meaningfully support interoperability and produce the desired functional outcome.

This document describes the integrations that can potentially be deployed at the same site. Wherever applicable, recommendations are given for the best practices flow for these integrations to work together. In addition, routing rule changes that you must make to suit the best practices flow are provided.

Additional changes must be made to the routing rules to interface these integrations with additional applications that are not supported by the integrations as delivered.

For more information about creating and using routing rules, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*, "Creating Oracle Mediator Routing Rules."

Before embarking on your interoperability project, ensure that you have installed the latest versions and patches for each product.

For more information about versions and patches, see the *Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations*.

1.2 Deployment Dependencies

Depending on the sequence in which pre-built integrations (or patches) are deployed, certain artifacts (for example, routing rules or DVMs) may be overwritten or returned to their original values by the deployment process.

For more information about deployment dependencies, see the *Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations*.

1.3 Functional Interoperability Matrix

Not all of the integrations that are marked as interoperable in the table below have chapters in this guide. Only those integrations that require manual changes (like routing rule changes) or that must follow certain guidelines to interoperate have chapters in this guide.

Table 1–1 provides a list of the Oracle AIA integrations along with the integrations with which they are functionally interoperable:

Table 1–1   Functionally Interoperable Pairs

Pre-Built Integration Name	Functionally Interoperable With:
Comms Order to Cash for Siebel, OSM, and BRM	▪ Comms Agent Assisted Billing Care: Siebel CRM - BRM ▪ Comms Revenue Accounting ▪ Customer MDM: Base ▪ Customer MDM: SBL CRM ▪ Customer MDM: Comms BRM ▪ Product MDM: Base ▪ Product MDM: SBL CRM ▪ Product MDM: Comms BRM
Comms Agent Assisted Billing Care: Siebel CRM - BRM	▪ Comms Order to Cash for Siebel, OSM, and BRM ▪ Comms Revenue Accounting ▪ Customer MDM: Base ▪ Customer MDM: SBL CRM ▪ Customer MDM: Comms BRM ▪ Product MDM: Base ▪ Product MDM: SBL CRM ▪ Product MDM: Comms BRM
Comms Revenue Accounting	▪ Comms Order to Cash for Siebel, OSM, and BRM ▪ Comms Agent Assisted Billing Care: Siebel CRM - BRM
Design to Release: Agile - EBS	▪ Order to Cash: Siebel CRM - EBS ▪ Product MDM: Base ▪ Product MDM: SBL CRM ▪ Product MDM: EBS ▪ Product MDM: Comms BRM **Note**: Design to Release: Agile - EBS and Product MDM products are co-deployable on a single Oracle AIA instance. They are not functionally interoperable.

Table 1–1   (Cont.) Functionally Interoperable Pairs

Pre-Built Integration Name	Functionally Interoperable With:
LSP Driver Mgmt: OTM - EBS	▪ LSP Financial Mgmt: OTM - EBS ▪ LSP Order Mgmt: OTM - EBS - Siebel CRM
LSP Financial Mgmt: OTM - EBS	▪ LSP Driver Mgmt: OTM - EBS ▪ LSP Order Mgmt: OTM - EBS - Siebel CRM
LSP Order Mgmt: OTM - EBS - Siebel CRM	▪ LSP Driver Mgmt: OTM - EBS ▪ LSP Financial Mgmt: OTM - EBS
Order to Cash: Siebel CRM - EBS	▪ Design to Release: Agile - EBS ▪ Customer MDM: Base ▪ Customer MDM: SBL CRM ▪ Customer MDM: EBS ▪ Customer MDM: Comms BRM ▪ Product MDM: Base ▪ Product MDM: SBL CRM ▪ Product MDM: Comms BRM
Customer MDM: Base	▪ Customer MDM: SBL CRM ▪ Customer MDM: Comms BRM ▪ Customer MDM: EBS ▪ Order to Cash: Siebel CRM - EBS ▪ Comms Agent Assisted Billing Care: Siebel CRM - BRM ▪ Comms Order to Cash for Siebel, OSM, and BRM
Customer MDM: SBL CRM	▪ Customer MDM: Base ▪ Customer MDM: Comms BRM ▪ Customer MDM: EBS ▪ Order to Cash: Siebel CRM - EBS ▪ Comms Agent Assisted Billing Care: Siebel CRM - BRM ▪ Comms Order to Cash for Siebel, OSM, and BRM
Customer MDM: EBS	▪ Customer MDM: Base ▪ Customer MDM: Siebel CRM ▪ Customer MDM: Comms BRM ▪ Order to Cash: Siebel CRM - EBS
Customer MDM: Comms BRM	▪ Customer MDM: Base ▪ Customer MDM: SBL CRM ▪ Customer MDM: EBS ▪ Comms Agent Assisted Billing Care: Siebel CRM - BRM ▪ Comms Order to Cash for Siebel, OSM, and BRM
Product MDM: Base	▪ Design to Release: Agile - EBS (co-deployable only) ▪ Product MDM: BRM ▪ Product MDM: SBL CRM ▪ Product MDM: EBS ▪ Order to Cash: Siebel CRM - EBS ▪ Comms Agent Assisted Billing Care: Siebel CRM - BRM ▪ Comms Order to Cash for Siebel, OSM, and BRM

Table 1–1   (Cont.) Functionally Interoperable Pairs

Pre-Built Integration Name	Functionally Interoperable With:
Product MDM: SBL CRM	- Design to Release: Agile - EBS (co-deployable only) - Product MDM: Base - Product MDM: BRM - Product MDM: EBS - Order to Cash: Siebel CRM - EBS - Comms Agent Assisted Billing Care: Siebel CRM - BRM - Comms Order to Cash for Siebel, OSM, and BRM
Product MDM: EBS	- Design to Release: Agile - EBS (co-deployable only) - Product MDM: Base - Product MDM: BRM - Product MDM: Siebel CRM - Order to Cash: Siebel CRM - EBS
Product MDM: Comms BRM	- Design to Release: Agile - EBS (co-deployable only) - Comms Order to Bill: Siebel CRM - BRM - Order to Cash: Siebel CRM - EBS - Comms Agent Assisted Billing Care: Siebel CRM - BRM - Comms Order to Cash for Siebel, OSM, and BRM

2

Design to Release: Agile - EBS with Order to Cash

This chapter discusses the interoperable pair of pre-built integrations: Agile Product Lifecycle Management Integration Pack for Oracle E-Business Suite: Design to Release (Design to Release: Agile - EBS) and Siebel CRM Integration Pack for Oracle Order Management: Order to Cash (Order to Cash: Siebel CRM - EBS). This chapter includes the routing rules that must be changed to enable interoperability.

This chapter includes the following sections:

The Design to Release: Agile - EBS integration supports the exchange of product information between Agile PLM and Oracle EBS. In the Order to Cash: Siebel CRM - EBS integration, product information is exchanged between Siebel Customer Relationship Management (Siebel CRM) and Oracle E-Business Suite (Oracle EBS). It is a one way feed from Oracle EBS to Siebel CRM.

For more information, see the *Oracle Application Integration Architecture Design to Release: Agile - EBS Implementation Guide* and the *Oracle Application Integration Architecture Siebel CRM Integration Pack for Oracle Order Management: Order to Cash*.

2.1 Product Sync Flows in Design to Release

The Design to Release: Agile - EBS integration enables and integrates the product development process between Design to Release: Agile - EBS and Oracle EBS. This integration supports different scenarios.

Change Order Release: during the product design phase, new products or parts are introduced or existing parts may go through design changes. When the authoring of a part's attributes and design information is complete and is ready for publishing to the manufacturing system, it is released by Change Orders. The Change Order Release Process consists of the New Part and Product Release (PREL) and the Product Design Modification flows of Design to Release: Agile - EBS. The release of a change order in the Design to Release: Agile - EBS integration system acts as a trigger for the synchronization of product design information with the ERP system.

For all items pushed to the ERP system, the system verifies whether the items exist, and whether they have the same revision number as the old item from the Agile system. If the item did not exist in ERP and is being released for the first time from PLM, it is created in ERP. If the item exists in ERP and the two systems were coordinated concerning item revision, the existing item is updated with new attribute data from Agile.

2.2 Product Sync Flows in Order to Cash

These are the product sync related flows:

- Sync Orderable Product from Oracle EBS to Siebel: When an item (product) is created or updated in Oracle EBS, it is synchronized to Siebel CRM.

- Request BOM Sync from Siebel to Oracle EBS: A product administrator in Siebel CRM requests the details of item relationships for the selected BOM from Oracle EBS to Siebel.

The details of the latest active version of the BOM and associated non-orderable components are synchronized from Oracle EBS to Siebel CRM as result of this request.

2.3 Best Practices for Flows

The following are the best practices for product (Item) and BOM synchronization among the various participating applications when the Design to Release: Agile - EBS integration and the Order to Cash: Siebel CRM - EBS support interoperability:

- The Design to Release: Agile - EBS integration manages the product life cycle.

- While the Design to Release: Agile - EBS integration is the system of record for item description, design, specs, and other information, the ERP system has more attributes and placeholders for information than the Design to Release: Agile - EBS system. The change order release must be updated in the ERP system.

- The update of *Item Attribute* information from Oracle EBS to Design to Release: Agile - EBS is done as a batch process.

- Item updates from Oracle EBS to Siebel CRM must be done using the Sync Item process.

- If needed, Siebel CRM should request the BOM from Oracle EBS.

2.4 Routing Rules Configuration

Routing rules changes are required to enable the interoperability of the Design to Release: Agile - EBS integration with the Order to Cash: Siebel CRM - EBS integration. The affected service-oriented architecture (SOA) composite service is the ItemEBSV2. The changes are delivered through patch 12314525. This patch should be applied to the Oracle AIA and Fusion Middleware (FMW) environment to re-deploy the ItemEBSV2 composite service with the new routing rules.

2.5 Post-Deployment Steps

Perform these steps after both integrations have been installed and deployed:

1. Log in to the Oracle AIA Console.

2. Navigate to **Setup**, **Systems**.

3. Verify that the value in the **System Type** column for the E-Business Suite system is set to *EBIZ* (all upper-case). If it is not set to *EBIZ*, change the value to *EBIZ* and save.

4. Navigate to the **AIA Configuration** tab, and click **Reload**.

3

Design to Release: Agile - EBS with Product MDM

This chapter discusses the co deployable group of pre-built integrations: Agile Product Lifecycle Management Integration Pack for Oracle E-Business Suite: Design to Release (Design to Release: Agile - EBS) and the Oracle Product Master Data Management set of pre-built integrations, namely Oracle Product Master Data Management Integration Base Pack (Product MDM: Base), Oracle Product Master Data Management Integration Option for Siebel CRM (Product MDM:SBL CRM), Oracle Product Master Data Management Integration Option for Oracle E-Business Suite (Product MDM:EBS) and Oracle Product Master Data Management Integration Option for Oracle Communications Billing and Revenue Management (Product MDM: Comms BRM).

This chapter includes the changes that must be done to enable co deployment.

This chapter includes the following sections:

- Section 3.1, "Product Sync Flows in Design to Release"
- Section 3.2, "Product Sync Flows in Product MDM"
- Section 3.3, "Best Practices for Flows"
- Section 3.4, "Configuration Steps"

The Design to Release: Agile - EBS integration supports the exchange of product information between Agile Product Lifecycle Management (Agile PLM) and Oracle E-Business Suite (Oracle EBS). In the Oracle Product Master Data Management (Product MDM) integration, product information is exchanged between Siebel Customer Relationship Management (Siebel CRM) and Oracle EBS. The Product MDM integration also supports the Oracle Product Hub (OPH) application in addition to Siebel CRM and Oracle EBS.

For more information, see the *Oracle Application Integration Architecture Design to Release: Agile - EBS Implementation Guide* and the *Oracle Application Integration Architecture Product Master Data Management Integration Pack Implementation Guide*

3.1 Product Sync Flows in Design to Release

The Design to Release: Agile - EBS integration enables and integrates the product development process between Design to Release: Agile and Oracle EBS. This integration supports different scenarios.

Change Order Release: During the product design phase, new products or parts are introduced, or existing parts may go through design changes. When the authoring of a part's attributes and design information is complete and is ready for publishing to the manufacturing system, it is released by Change Orders. The Change Order Release

process consists of the New Part and Product Release (PREL) and the Product Design Modification flows of Design to Release: Agile - EBS. The release of a change order in the Design to Release: Agile - EBS integration system acts as a trigger for the synchronization of product design information with the Enterprise Resource Planning (ERP) system. For all items pushed to the ERP system, the system verifies whether the items exist, and whether they have the same revision number as the old item from the Agile system. If the item did not exist in ERP and is being released for the first time from PLM, it is created in ERP. If the item exists in ERP and the two systems were coordinated concerning item revision, the existing item is updated with new attribute data from Agile.

3.2 Product Sync Flows in Product MDM

OPH is a single consolidated system for defining products (items), Bill of Materials (BOMs, or item structures), and more product related features.

The product MDM integration provides the following flows:

- Synchronize Product Classes from the OPH to Siebel CRM

- Synchronize Domain Values from the OPH to Siebel CRM

- Synchronize Items and BOM from the OPH to Siebel CRM

- Synchronize Items and BOM from the OPH to Oracle EBS

3.3 Best Practices for Flows

Agile Design to Release and Product MDM products are co-deployable on a single Oracle Application Integration Architecture (Oracle AIA) instance. They are not functionally interoperable. The configuration steps, (see Section 3.4), ensure that both the pre-built integrations work independent of each other in a single Fusion Middleware instance.

3.4 Configuration Steps

The following configuration steps must be performed to make Agile Design to Release and Product MDM products co-deployable on a single Oracle Application Integration Architecture (Oracle AIA) instance. There may be deployment sequence considerations.

> **Note:** Define all the references for itemEBSV2 and ItemResponseEBSV2 before using them in the routing rules.

To enable co-deployment:

Agile Design to Release and Product MDM products are co-deployable on a single Oracle AIA instance. They are not functionally interoperable.

1. Modify the routing rules for `ItemEBSV2` for SyncItemList operation as shown in Table 3–1 and then redeploy it in the Oracle AIA Fusion Middleware instance.

   `ItemEBSV2` is located under *AIA_HOME/pips/AgileToEbiz/EBS/Item/R12*.

Table 3–1   Routing Rules Configuration for ItemEBSV2

Operation	Filter Condition	Type	Service Invoked	Description of Routing Rules
SyncItemList	contains($in.SyncItemListEBM/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode,'PRODUCTION') and (aia:getSystemType(normalize-space($in.SyncItemListEBM/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:Sender/corecom:ID/text())) ='AGILE') or (aia:getSystemType(normalize-space($in.SyncItemListEBM/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:Target/corecom:ID/text()))='EBIZ')) and (aia:getSystemType(normalize-space($in.SyncItemListEBM/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:Sender/corecom:ID/text())) !='EBIZ')	Sequential	SyncItemListEbizProvABCSImpl:: SyncItemList	Default routing rule in Agile pre-built integration
SyncItemList	(aia:getServiceProperty('{http://xmlns.oracle.com/EnterpriseServices/Core/Item/V2}ItemEBSV2','Routing.ActiveRuleset',false())='DEFAULT' and count(normalize-space($in.SyncItemListEBM/ebo:SyncItemListEBM/ebo:DataArea)) > 0 and (normalize-space($in.SyncItemListEBM/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text())='PRODUCTION' or not (normalize-space($in.SyncItemListEBM/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text()))) and (aia:getSystemType(normalize-space($in.SyncItemListEBM/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:Target/corecom:ID/text()))='SIEBEL') and (aia:getSystemType(normalize-space($in.SyncItemListEBM/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:Sender/corecom:ID/text())) !='SIEBEL'))	Sequential	SyncProductSiebelProvABCSImpl:: SyncItem	The routing rule in the Agile pre-built integration by default routes the messages to Oracle EBS. This new routing rule routes the messages to Siebel CRM used by Product MDM pre-built integration.
SyncItemList	(aia:getServiceProperty('ebsItemEBSV2','Routing.ActiveRuleset',false())='DEFAULT' and count(normalize-space($in.SyncItemListEBM/ebo:SyncItemListEBM/ebo:DataArea)) > 0 and(normalize-space($in.SyncItemListEBM/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text())='PRODUCTION' or not (normalize-space($in.SyncItemListEBM/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text())))) and (aia:getSystemType(normalize-space($in.SyncItemListEBM/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:Target/corecom:ID/text()))='BRM') and (aia:getSystemType(normalize-space($in.SyncItemListEBM/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:Sender/corecom:ID/text())) !='SIEBEL'))	Sequential	SyncItemListBRMProvABCSImpl:: SyncItemList	The routing rule in the Agile pre-built integration by default routes the messages to Oracle EBS. This new routing rule routes the messages to Oracle BRM used by the Product MDM pre-built integration.

> **Caution:** Using the Oracle Billing and Revenue Management (Oracle BRM) application is optional in the Product MDM pre-built integration. Therefore, adding the routing rule for `SyncItemListBRMProvABCSImpl` must only be done when Oracle BRM is used as the edge application in the Product MDM pre-built integration.

2. Modify the routing rules for `ItemResponseEBSV2` for the SyncItemListResponse operation as shown in Table 3–2 and then redeploy it in the Oracle AIA Fusion Middleware instance.

   `ItemResponseEBSV2` is located under *AIA_HOME/pips/AgileToEbiz/EBS/Item/R12*.

Table 3–2 Routing Rules Configuration for ItemResponseEBSV2

Operation	Filter Condition	Type	Service Invoked	Description of Routing Rules
SyncItemList Response	contains($in.SyncItemListResponseEBM/ebo:SyncIt emListResponseEBM/corecom:EBMHeader/coreco m:MessageProcessingInstruction/corecom:Environ mentCode,'PRODUCTION') and (aia:getSystemType(normalize-space($in.SyncItemL istResponseEBM/ebo:SyncItemListResponseEBM/c orecom:EBMHeader/corecom:Sender/corecom:ID/ text()))='EBIZ') and ($in.SyncItemListResponseEBM/ebo:SyncItemListR esponseEBM/corecom:EBMHeader/corecom:Targe t/corecom:ApplicationTypeCode='AGILE')	Sequential	SyncItemListAgileReqA BCSImpl:: receive	Default routing rule in Agile pre-built integration
SyncItemList Response	contains($in.SyncItemListResponseEBM/ebo:SyncIt emListResponseEBM/corecom:EBMHeader/coreco m:MessageProcessingInstruction/corecom:Environ mentCode,'PRODUCTION') and (aia:getSystemType(normalize-space($in.SyncItemL istResponseEBM/ebo:SyncItemListResponseEBM/c orecom:EBMHeader/corecom:Sender/corecom:ID/ text()))='EBIZ' or aia:getSystemType(normalize-space($in.SyncItemLi stResponseEBM/ebo:SyncItemListResponseEBM/c orecom:EBMHeader/corecom:Sender/corecom:ID/ text()))='SIEBEL') and ($in.SyncItemListResponseEBM/ebo:SyncItemListR esponseEBM/corecom:EBMHeader/corecom:Targe t/corecom:ApplicationTypeCode='PIM')	Sequential	SyncItemListPIMReqAB CSImpl:: ReceiveSyncItemListRes ponse	The routing rule in the Agile pre-built integration by default routes the messages to Agile. This new routing rule routes the messages to PIM used by Product MDM pre-built integration.

3. Make sure that the application type for the Oracle EBS application is set as *EBIZ_ 01* in the AIA console, as shown in Figure 3–1.

   This is required for running the Variant Management flow in Agile Design to Release pre-built integration. When the Agile Design to Release pre-built integration is installed on top of Product MDM, the application type remains as *EBIZ* and causes failures when performing the lookups on the Domain Value Maps (DVMs).

Figure 3–1 AIA Console - Systems
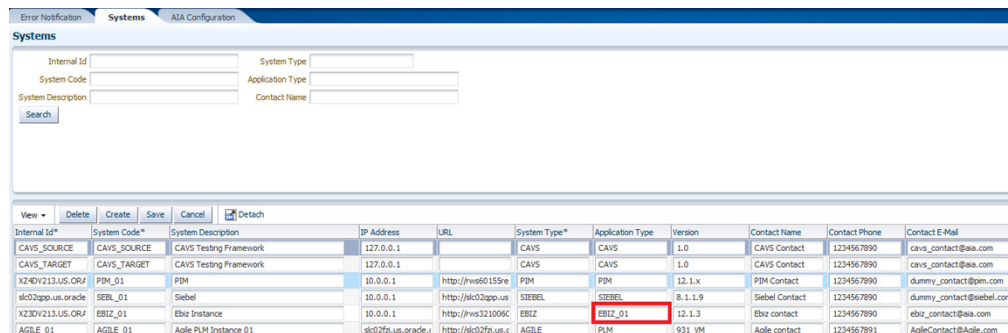
4. Verify and correct the *AIAConfigurationProperties.xml* file, as shown in Example 3–1.

Installing Product MDM on top of the Agile Design to Release pre-built integration, may alter the properties listed in Example 3–1. Verify that all the properties listed here are in the *AIAConfigurationProperties.xml* file.

> **Note:** The values for the properties listed in Example 3–1 are user-specific and therefore may be altered.

Example 3–1 AIAConfigurationProperties

```
<ModuleConfiguration moduleName="Ebiz">
          <Property name="EBIZ_01.SERVER_TIMEZONE">-08:00</Property>
  <!-- Added these properties manually as these are missing in co deployment -->
 <Property name="FIXED_ORG_FOR_INBOUND_DATA_TRANSFER"/>
<Property name="IS_PIM_INSTALLED">true</Property>
<Property name="DEFAULT_MASTER_ORG"/>
<Property name="USER">operations</Property>
<Property name="RESPONSIBILITY">System Administrator</Property>
<!-- The property EBIZ_01.SERVER_GMT_OFFSET will be deprecated in 2.5,
and will be replaced with EBIZ_01.SERVER_TIMEZONE-->
<Property name="EBIZ_01.SERVER_GMT_OFFSET">+00:00</Property>
<!-- Please refer Ebiz profile SERVER_TIMEZONE_ID to configure this property, e.g.
'-07:00' or 'GMT-07:00' or 'PST' -->
<Property name="EBIZ_01.SERVER_TIMEZONE">GMT-08:00</Property>
<Property name="EBIZ_01.LANG">ENG</Property>
<Property name="PROJECTS_RESPONSIBILITY">Projects, Vision Services
(USA)</Property>
        </ModuleConfiguration>
```

After modifying the *AIAConfigurationProperties.xml* file, you must ensure that the changes are also reflected in Oracle Service-Oriented Architecture Metadata Services (Oracle SOA MDS) repository.

4

Customer MDM with Order to Cash

This chapter discusses the interoperable group of pre-built integrations: Oracle Customer Master Data Management Integration Base Pack (Customer MDM: Base), Oracle Customer Master Data Management Integration Option for Siebel CRM (Customer MDM: SBL CRM), Oracle Customer Master Data Management Integration Option for Oracle E-Business Suite (Customer MDM: EBS) and Siebel CRM Integration Pack for Oracle Order Management: Order to Cash (Order to Cash: Siebel CRM - EBS). This chapter includes the routing rules that must be changed to enable interoperability.

This chapter includes the following sections:

- Section 4.1, "Customer Creates or Updates in Siebel CRM"
- Section 4.2, "Customer Creates or Updates in Oracle EBS"
- Section 4.3, "Merge Process"
- Section 4.4, "Best Practices for Flows"
- Section 4.5, "Interoperability Routing Rules Deployment"
- Section 4.6, "Interoperability Routing Rules Undeployment"

In the Customer MDM integration, customer information is synchronized from Oracle Customer Hub (OCH) to Siebel Customer Relationship Management (Siebel CRM) and Oracle E-Business Suite (Oracle EBS). In the Order to Cash: Siebel CRM - EBS integration, customer information is synchronized bi-directionally between Siebel CRM and Oracle EBS.

For more information, see the *Oracle Application Integration Architecture Customer Master Data Management Integration Pack Implementation Guide* and the *Oracle Application Integration Architecture Siebel CRM Integration Pack for Oracle Order Management: Order to Cash*.

4.1 Customer Creates or Updates in Siebel CRM

These are the recommendations for Siebel CRM:

- Customers must be created in Siebel CRM using the match-fetch flows.

  Once created, customers should be routed to Oracle Customer Hub instead of being directly routed to the target applications, in this case, Oracle EBS. Oracle Customer Hub must be configured to publish these created customers to the target application.

- The order process integration in the Order to Cash: Siebel CRM - EBS integration has a flow that queries the account associated to the sales order from Siebel CRM.

This flow further creates account sites and site-uses in the same business unit as the sales order in Oracle EBS. This customer synchronization flow remains the same and, only as part of this flow, new accounts are created in the Oracle EBS fulfillment system.

■ When Oracle Customer Hub publishes messages to target systems, if an implementation requires that only updates to the existing accounts should be sent to the Oracle EBS fulfillment system, then a transform that filters only updates can be applied.

Applying this filter enables implementations to avoid prospect records (without any orders) from being synced with fulfillment systems. This is documented in Section 4.5, "Interoperability Routing Rules Deployment".

■ There could be a situation where an account may be queued or identified as a duplicate in Oracle Customer Hub which is handled by the synchronize account or merge Oracle Customer Hub ABCS service.

However, if an order were submitted using this account, the redundant customer synchronization from order process would synchronize the customer to Oracle EBS and ensure order submission.

4.2 Customer Creates or Updates in Oracle EBS

This is the recommendation for Oracle EBS:

■ When customer records are created or updated in Oracle EBS, they should be routed to Oracle Customer Hub.

These transactions should not be synchronized directly to target applications (Siebel CRM). The updated records should be published from Oracle Customer Hub to target applications (Siebel CRM).

4.3 Merge Process

These are the recommendations for using the merge process:

■ Account or contact merges can take place only in Oracle Customer Hub.

■ Merge messages from Oracle Customer Hub are routed to all target applications registered within Oracle Customer Hub.

■ In Order to Cash, the merge flow is from Oracle E-Business Suite to Siebel and is initiated manually. Merges should not be done in Oracle E-Business Suite.

■ The Order to Cash: Siebel CRM - EBS integration functionality of party or account merge in Oracle EBS is not supported in interoperability with Customer MDM integration merge functionality.

4.4 Best Practices for Flows

For Order to Cash and Customer MDM integration interoperability, these flows are recommended for the customer account synchronization among the participating applications:

■ Customer creates or updates must be done in Siebel CRM.

■ Customer creates or updates must be done in Oracle E-Business Suite.

■ The merge process must be done in Oracle Customer Hub.

4.5 Interoperability Routing Rules Deployment

Changes are required to the routing rules that were deployed out-of-the-box for the Customer MDM integration or the Order to Cash: Siebel CRM – EBS integration. The sequence in which the integrations were deployed does not hold any significance here. The modified rules have to be deployed, following the instructions in this section.

For more information about creating and using routing rules, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*, "Creating Oracle Mediator Routing Rules."

Routing rules in the following Mediator services are modified to support interoperability:

- CustomerPartyEBSV2
- CustomerPartyResponseEBSV2
- CustomerPartyOrchestrationEBSV2
- CustomerPartyResponseOrchestrationEBSV2

To deploy the routing rules for the Interoperability scenarios to the Fusion Middleware server:

1. Navigate to <AIA_HOME>/aia_instances/<AIA_InstanceName>/bin and run the command source `aiaenv.sh` for Linux based systems and `aiaenv.bat` for Microsoft Windows to configure the environment.

2. Run the command for your platform.

   Table 4–1 shows the deployment commands for interoperability routing rules:

Table 4–1   Deployment Commands for Interoperability Routing Rules

Platform	Deployment Command
Linux x86	sh $AIA_HOME/pips/Interoperability/MDMCustomer_ OrderToCash/DeploymentPlans/deployMDMCustomerEBS.sh
Solaris SPARC (64-bit)	
IBM AIX Based Systems (64-bit).	
HP-UX 11i (64 bit)	
Microsoft Windows (32-bit)	%AIA_HOME%\pips\Interoperability\MDMCustomer_ OrderToCash\DeploymentPlans\deployMDMCustomerEBS.bat

3. Review the log file in the location specified in the command or at the default location <AIA_Instance>/logs/MDMCustomerEBSDP_Deployments_ YYYY-MM-DD_HH-MI-SS.log to verify successful deployment.

4.6 Interoperability Routing Rules Undeployment

To undeploy the routing rules from Fusion Middleware Server:

1. Navigate to <AIA_Instance>/bin and run the command source `aiaenv.sh` for Linux based systems and `aiaenv.bat` for Microsoft Windows to configure the environment.

2. Run the command for your platform.

   Table 4–2 shows the undeployment commands for routing rules:

Table 4–2   Undeployment Commands for Interoperability Routing Rules

Platform	Undeployment Commands
Linux x86 Solaris SPARC (64-bit) IBM AIX Based Systems (64-bit). HP-UX 11i (64 bit)	sh $AIA_HOME/pips/Interoperability/MDMCustomer_ OrderToCash/DeploymentPlans/undeployMDMCustomerEBS.sh
Microsoft Windows (32-bit)	%AIA_HOME%\pips\Interoperability\MDMCustomer_ OrderToCash\DeploymentPlans\undeployMDMCustomerEBS.bat

3. Restart the SOA server to clear out the cached content.

Performing these steps completely undeploys the following Mediator services:

- CustomerPartyEBSV2

- CustomerPartyResponseEBSV2

- CustomerPartyOrchestrationEBSV2

- CustomerPartyOrchestrationResponseEBSV2

5

Product MDM with Order to Cash

This chapter discusses the interoperable group of pre-built integrations: Oracle Product Master Data Management Integration Base Pack (Product MDM: Base), Oracle Product Master Data Management Integration Option for Siebel CRM (Product MDM: SBL CRM), Oracle Product Master Data Management Integration Option for Oracle Communications Billing and Revenue Management (Product MDM: Comms BRM) and Siebel CRM Integration Pack for Oracle Order Management: Order to Cash (Order to Cash: Siebel CRM - EBS). This chapter includes the routing rules that must be changed to enable interoperability.

This chapter includes the following sections:

- Section 5.1, "Product Flows in Product MDM"
- Section 5.2, "Product Flows in Order to Cash"
- Section 5.3, "Best Practices for Flows"
- Section 5.4, "Configuration Steps"

In both the Product MDM integration and the Order to Cash integration, product information is synchronized between Siebel Customer Relationship Management (Siebel CRM) and Oracle E-Business Suite (Oracle EBS). The Product MDM integration also supports the Oracle Product Hub application in addition to Siebel CRM and Oracle EBS.

For more information, see the *Oracle Application Integration Architecture Product Master Data Management Integration Pack Implementation Guide* and the *Oracle Application Integration Architecture Siebel CRM Integration Pack for Oracle Order Management: Order to Cash*.

5.1 Product Flows in Product MDM

The Oracle Product Hub (OPH) is a single consolidated system for defining products (items), Bill of Materials (BOMs, or item structures), and more product related features. The flows provided by the Product MDM integration are:

- Synchronize Product Classes from the OPH to Siebel CRM.
- Synchronize Domain Values from the OPH to Siebel CRM.
- Synchronize Items and BOM from the OPH to Siebel CRM.
- Synchronize Items and BOM from the OPH to Oracle E-Business Suite.

5.2 Product Flows in Order to Cash

The flows provided by the Order to Cash integration are:

- Sync Orderable Product from Oracle EBS to Siebel CRM.

  When an item (product) is created or updated in Oracle EBS, it is synchronized to Siebel CRM.

- Request Bill of Material (BOM) Sync from Siebel to Oracle EBS.

  A product administrator in Siebel CRM requests the details of item relationships for the selected BOM from Oracle EBS. As a result of this request, the details of the latest active version of the BOM and the associated non-orderable components are synchronized from Oracle EBS to Siebel CRM.

5.3 Best Practices for Flows

In the Product MDM integration, product information is synchronized from OPH to Siebel CRM and Oracle EBS. In the Order to Cash integration, product information is synchronized uni-directionally between Oracle EBS and Siebel CRM.

For more information, see the *Oracle Application Integration Architecture Product Master Data Management Integration Pack Implementation Guide* and the *Oracle Application Integration Architecture Siebel CRM Integration Pack for Oracle Order Management: Order to Cash*.

When both the Product MDM integration and the Order to Cash integration are deployed, the following best practices flows are recommended for the product and bill of materials (BOM) synchronization among the various participating applications.

- Products are created and updated in OPH only, so the AIA configuration property 'Product.Source.PIP' is always MDM.

- Synchronize products from the Product MDM integration to Siebel CRM and Oracle EBS.

- Sync BOMs from the Product MDM integration to Siebel CRM and Oracle EBS.

- Request BOM synchronization from Siebel to Oracle EBS.

- Synchronize Product Classes from the Product MDM integration to Siebel.

- Synchronize Domain Values from the Product MDM integration to Siebel.

Even though the BOM is synchronized from the Product MDM integration, this BOM does not contain the Oracle EBS identifiers required to invoke Oracle Product Configurator from Siebel CRM. The product administrator in Siebel CRM must request the details of the product structure for the selected BOM from Oracle EBS. This synchronization provides the Oracle EBS identifiers in Siebel, so that the Oracle Product Configurator can be successfully invoked.

5.4 Configuration Steps

This section describes the routing rule and AIA configuration property changes that must be made to enable interoperability. There may be deployment sequence considerations.

This section includes the following topics:

- Section 5.4.1, "Routing Rules Configuration"
- Section 5.4.2, "AIA Configuration Properties"

For more information about install sequence considerations, see Chapter 1.2, "Deployment Dependencies". For more information about using and extending routing rules, see the *Oracle Enterprise Service Bus Developer's Guide*, "Creating Routing Services and Routing Rules".

5.4.1 Routing Rules Configuration

Configure the ItemEBSV2 routing rule as described in Section 5.4.1.1, "For ItemEBSV2".

5.4.1.1 For ItemEBSV2

A new filter condition for the routing rule to the SyncProductSiebelProvABCSImpl must be added in the ItemEBSV2 service.

Table 5–1 shows the filter condition:

Table 5–1 Routing Rules for ItemEBSV2

Operation	Filter Condition	Service Invoked	Description of Routing Rules

SyncItemList	`(aia:getServiceProperty('{http://xmlns.oracle.co`
	`m/EnterpriseServices/Core/Item/V2}ItemEBSV2',`
	`'Routing.ActiveRuleset',false())='DEFAULT'`
	` and`
	`count(normalize-space($in.SyncItemListEBM/ebo:Sy`
	`ncItemListEBM/ebo:DataArea))`
	` > 0 and`
	`(normalize-space($in.SyncItemListEBM/ebo:SyncIte`
	`mListEBM/corecom:`
	`EBMHeader/corecom:MessageProcessingInstruction/c`
	`orecom:EnvironmentCode/text`
	`())='PRODUCTION' or not`
	`(normalize-space($in.SyncItemListEBM/ebo:SyncIte`
	`mListEBM/corecom:`
	`EBMHeader/corecom:MessageProcessingInstruction/c`
	`orecom:`
	`EnvironmentCode/text()))) and`
	`(xref:lookupXRef('oramds:/apps/AIAMetaData/xref/`
	`INVENTORY_LOCATION_`
	`ID.xref','COMMON',$in.SyncItemListEBM/ebo:SyncIt`
	`emListEBM/ebo:DataArea[1]/ebo:SyncItemList/corec`
	`omEBO:`
	`InventoryLocation/corecom:Identification/corecom`
	`:BusinessComponentID,'SEBL_01',false())) `
	` and`
	`((normalize-space($in.SyncItemListEBM/ebo:SyncIt`
	`emListEBM/corecom:EBMHeader/corecom:`
	`Target/corecom:ApplicationTypeCode/text())='SIEB`
	`EL' or`
	`not(normalize-space($in.SyncItemListEBM/ebo:Sync`
	`ItemListEBM/corecom:`
	`EBMHeader/corecom:Target/corecom:ID/text()))) or`
	`aia:getSystemType(normalize-space`
	`($in.SyncItemListEBM/ebo:SyncItemListEBM/corecom`
	`:EBMHeader/corecom:Target/corecom:ID/text`
	`()))='SIEBEL') and`
	`(aia:getSystemType(normalize-space`
	`($in.SyncItemListEBM/ebo:SyncItemListEBM/corecom`
	`:EBMHeader/corecom:Sender/corecom:ID/text()))`
	`!='SIEBEL'))`
		SyncProductSiebel ProvABCSImpl_1_ 0::SyncItem	The routing rule in the MDM integration by default only routes product sync messages created by PIM to Siebel.
			
			The additional conditions in the filter documented here, ensure that "*-items" created in Ebiz, are synced to Siebel since those types of items are mastered in E-Business Suite in the Order to Cash integration scenario.

5.4.2 AIA Configuration Properties

Set the following properties in the *AIAConfigurationProperties.*xml file:

1. Set the AIAConfiguration property **PUBLISH_ALL_PRODUCT** for the service **QueryItemCompositionListEbizProvABCSImpl** to *Y*.

   Setting this property allows synchronization of all the Oracle EBS identifiers in Siebel.

2. Set the AIAConfiguration property **Product.Source.PIP** for both the **InterfaceSyncProductStructureEBF** and the **SyncProductSiebelProvABCSImpl** services to *MDM*.

3. Set the AIAConfiguration property **COMPLEX_PRODUCT_RELEASE_FLAG, SIMPLE_PRODUCT_RELEASE_FLAG** for the service **SyncItemCompositionListSiebelProvABCSImpl** to *Y*.

6

Customer MDM with Communications Order to Cash

This chapter discusses the interoperable group of pre-built integrations: Oracle Customer Master Data Management Integration Pack (Customer MDM: Base), Oracle Customer Master Data Management Integration Option for Siebel CRM (Customer MDM: SBL CRM), Oracle Customer Master Data Management Integration Option for Oracle Communications Billing and Revenue Management (Customer MDM: Comms BRM) and Oracle Communications Order to Cash - Siebel CRM Option and Oracle BRM Option (Comms Order to Cash). This chapter includes the routing rules that must be changed to enable interoperability.

This chapter includes the following sections:

- Section 6.1, "Customer Flows in Customer MDM"
- Section 6.2, "Customer Flows in Comms Order to Cash"
- Section 6.3, "Best Practices for Flows"
- Section 6.4, "Merge Process in Customer MDM"
- Section 6.5, "Merge Process in Comms Order to Cash"
- Section 6.6, "Best Practices for Using the Merge"
- Section 6.7, "Routing Rules Configuration"

6.1 Customer Flows in Customer MDM

Upon creation of a new customer in Siebel CRM, the Oracle Customer MDM integration sends the customer data to the Oracle Customer Hub. The customer data then goes through a sanitation process within the Oracle Customer Hub where it is cleansed, standardized, matched, and linked to other existing customer entries and possibly enriched. The updated customer data is then returned to Siebel and published to other subscribing applications.

Every time a customer is created in Siebel CRM, that information is published to the Oracle Customer Hub. Even prospects (customers that decide not to go ahead with an order) are published to Oracle Customer Hub. However, prospects are not published to Oracle Billing and Revenue Management (Oracle BRM). Prospect data is of no use to a billing system, and its presence creates a performance overhead.

6.2 Customer Flows in Comms Order to Cash

In the Comms Order to Cash integration, upon submission of the order, customer data references are published with the order.

To avoid creating prospects in Oracle BRM, the creation of customer data in Oracle BRM only takes place through an order submission, as part of the Create Customer Data in Billing flow. In this integration point, Oracle Application Integration Architecture (Oracle AIA) detects if the customer is new to Oracle BRM (missing cross references) in which case the customer data is queried from Siebel and created in Oracle BRM.

The updates made by Oracle Customer Hub to the customer data through cleansing, standardization, and enrichment must be applied to the data in Oracle BRM. When Oracle Customer Hub publishes the cleansed data to other systems, it also publishes the same data to Oracle BRM. The routing rules check if the published customer data is used in Oracle BRM by checking the cross-references. If so, the message is passed to provider system and it applies the data received from the Oracle Customer Hub as an update to the existing customer data, or else it discards it.

Figure 6–1 shows the create customer data flows:

Figure 6–1 Creating Customer Data Across MDM Customer and AIACOM Flows
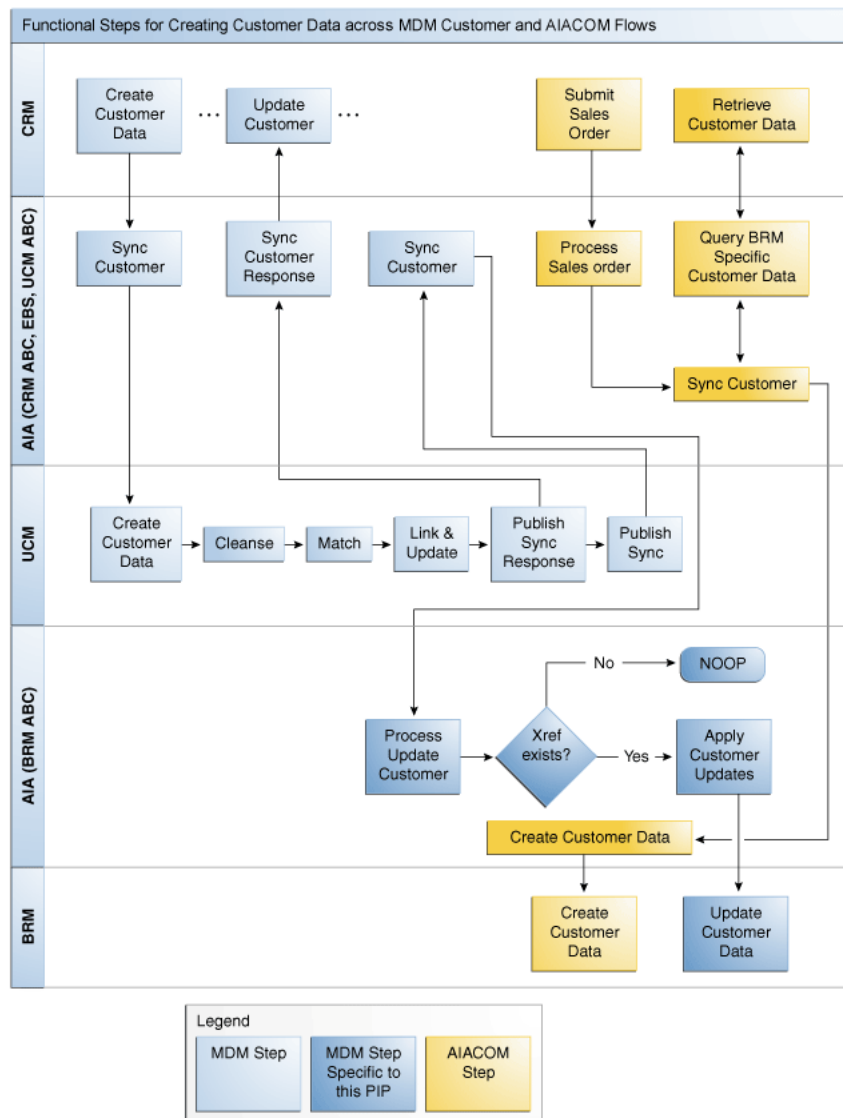
For more information, see the *Oracle Application Integration Architecture Customer Master Data Management Integration Pack Implementation Guide* and the *Oracle Application Integration Architecture Oracle Communication Order to Cash Integration Pack Implementation Guide for Siebel CRM, Oracle Order and Service Management, and Oracle Billing and Revenue Management*.

6.3 Best Practices for Flows

These are the recommendations:

- Customers must be created in Siebel CRM using the match-fetch flows.

  Once created, customers should be routed to Oracle Customer Hub instead of being directly routed to the target applications, in this case, Oracle BRM. The Oracle Customer Hub should be configured to publish the newly created customers to the target application.

- Publishing from Oracle Customer Hub to Oracle BRM must be done in batch.

  How often the batch is run depends on the specific deployment conditions. It can be set from near-realtime sync (minimum delay in the batch) to a daily batch. Factors that must be considered in determining the frequency of the batch publishing include: average duration of the order capture process, frequency of bill cycles, performance impact, and possible requirements for having near-realtime updated customer data in the billing system.

- When interoperating with applications that do not support a merge process, if you used the merge in Oracle Customer Hub, it is recommended that you turn off the publish to the participating applications.

  For more information, see Section 6.6.1, "To Disable the Publish Process".

- To enable functional interoperability, the AIA configuration property ProcessUpdateEventsOnly must be set to *false* when both of these integrations are deployed. The default value for this property is *false*, but certain flows require that this property is set to *true*. You must change this configuration property based on the flows being used:

  - If customers are only using the Create Customer and the Update Customer flows in the Customer MDM integration, then the ProcessUpdateEventsOnly property must be *false*.

  - If customers are only using Comms Order to Cash and Comms Agent-Assisted Billing Care, then the value of the flag must be set to *true*. This is an optimization guideline, because in the Comms flows, customer creation is enabled through the order flows. Consequently, setting the value to *true* ensures that only update events are propagated outward from the Siebel customer sync requestor service.

  - However, if customers are using both Customer MDM and Comms Order to Cash or Agent-Assisted Billing Care, then the property must be *false* to enable the creates from Siebel CRM to be propagated to the Oracle Customer Hub.

- A race condition may occur between an account create or update coming from Siebel CRM to Oracle BRM and the updates coming from OCH to Oracle BRM.

  - If customer data is created in Siebel CRM and published to Oracle BRM, the updates to OCH are applied after the customer data has been created.

  - If customer data is updated in Siebel CRM and published to OCH and Oracle BRM, OCH also publishes to Oracle BRM after performing cleansing and

de-duping. In this case, changes to Oracle BRM are applied by OCH only after the Siebel CRM updates have been applied to OCH.

– This race condition is a rare event as it takes place only when the update from OCH reaches Oracle BRM before it reaches Siebel CRM, and before the AIACOM query from Siebel CRM.

For more information about the race condition, see the *Oracle Application Integration Architecture Customer Master Data Management Integration Pack Implementation Guide*, "Oracle Customer MDM Integration Base Pack", Synchronization of Organizations and Persons from Oracle Customer Hub to Oracle BRM".

Note: The status of any entity is special compared to other fields of that entity. Account status or Contact status changes from *valid* to *invalid* or from *active* to *inactive*, have implications in various systems such as billing or CRM or ERP.

If you must update status in systems other than OCH, ensure the survivorship rules are defined properly in OCH, so that this status update is not rejected by OCH. If survivorship rules are not configured properly, then OCH rejects the update and reverses the status back in CRM (or any other source application). This is true for any updates to an account or a contact.

When the Oracle Communications Order To Cash Siebel CRM and BRM options are deployed along with the Customer MDM Base Pack and BRM options, and the account status sync is enabled (in Comms O2C), then there is a chance that by the time OCH rejects the update from CRM, BRM may have been updated with the status information from CRM directly.

To avoid this mismatch between Siebel CRM and BRM, if OCH is the master for account status, Oracle recommends that customers put business process controls in place to disallow changes to account status in Siebel CRM.

Alternatively, if status must be updated in any source system, then you must ensure that survivorship rules are configured in OCH to allow updates from any source system to be accepted by OCH.

For more information about the implications of enabling status sync to BRM see the *Oracle Application Integration Architecture Oracle Communication Order to Cash Integration Pack Implementation Guide for Siebel CRM, Oracle Order and Service Management, and Oracle Billing and Revenue Management*, "Account Status Synchronization Methodology."

6.4 Merge Process in Customer MDM

In the Customer MDM integration, the merge functionality is a core function that allows a data steward to effectively determine a survivor record among several duplicate records, or to create a record to consolidate several duplicate or similar records. For this process, Oracle Customer Hub publishes the results of the merge by providing the **survivor** record and one or many **victim** records to the source systems. It is up to the source systems to determine whether they can or should consume this message in their own records. Although some participating applications have the ability to request a merge operation, only customer merges executed within Oracle Customer Hub are supported for this release.

6.5 Merge Process in Comms Order to Cash

Siebel CRM has a service to merge records and re-parent the associated entities to reflect the merge performed by Oracle Customer Hub. Unfortunately, Oracle BRM does not offer a service to support this feature. Obviously merging accounts or contacts in Siebel CRM without merging the corresponding accounts or contacts in Oracle BRM is not a feasible option, since it is not possible to retrieve invoice details for victim accounts or submit change orders related to assets parented to victim accounts.

Clients can develop ad-hoc procedures and tools so that administrators can merge customer data in Siebel CRM and in Oracle BRM to be consistent with the merge data in Oracle Customer Hub. This is a custom solution not supported by the delivered flows.

6.6 Best Practices for Using the Merge

The following are the recommendations for using the merge process:

- Account or contact merges can take place only in Oracle Customer Hub.

- Merge messages from Oracle Customer Hub are routed to all target applications registered within Oracle Customer Hub, including Oracle BRM.

- Merges are not done in Oracle BRM when a merge message is received.

  The Oracle BRM provider checks the cross-reference for the survivor customer data and if it exists, updates the survivor customer data. The same occurs for a sync message on the survivor.

6.6.1 To Disable the Publish Process

When interoperating with applications that do not support a merge process, if you used the merge in Oracle Customer Hub, it is recommended that you turn off the publish to the participating applications.

Therefore, for Comms Order to Cash processes that interoperate with Customer MDM processes, when a merge happens in Oracle Customer Hub, it should not be published to the Siebel CRM or Oracle BRM instances that participate in the Comms Order to Cash integration. If the merge is published to Siebel, and the cross-references are updated, then the change order and customer update flows break.

Perform the following steps to turn off the publish after the merge:

1. Query the Oracle Customer Hub Process Merge Request workflow from **Siebel tools**.

2. Click **Revise** to revise the workflow.

3. Set the value of the process property **EnablePubSub** = **False**.

4. Deploy the workflow.

For more information, see the Siebel Book Shelf: **Application Development**, **Siebel Business Process Framework**, **Workflow Guide**.

6.7 Routing Rules Configuration

For Customer MDM and Comms Order to Cash interoperability, configuration depends on the order of deployment.

■ If the Customer MDM integration is deployed first, and then the Comms Order to Cash integration is deployed, no routing rules changes or additions are required.

■ If the Comms Order to Cash integration is deployed first, and then the Customer MDM integration is deployed:

1. Use the deployment plan *MDMCustomerEBSDP*.xml at $AIA_ HOME/pips/MDMCustomer/DeploymentPlans to redeploy CustomerPartyEBSV2 with overwrite flag set to true by updating the deployment plan as follows:

```
<Composite compositeName="CustomerPartyEBSV2"
compositedir="${AIA_
HOME}/pips/MDMCustomer/EBS/CustomerParty/CustomerPartyEBSV
2" revision="1.0" overwrite="true"
wlserver="pips.MDMCustomerBase" action="deploy" />
```

2. Comment out the rest of the composites and run the *MDMCustomerEBSDP*.xml deployment plan.

7

Customer MDM with Communications Agent Assisted Billing Care

This chapter discusses the interoperable group of pre-built integrations: Oracle Customer Master Data Management Integration Base Pack (Customer MDM: Base), Oracle Customer Master Data Management Integration Option for Siebel CRM (Customer MDM: SBL CRM), Oracle Customer Master Data Management Integration Option for Oracle Communications Billing and Revenue Management (Customer MDM: Comms BRM) and Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care (Comms Agent Assisted Billing Care). This chapter includes the routing rules that must be changed to enable interoperability.

This chapter contains the following sections:

- Section 7.1, "Customer Flows in Customer MDM"

- Section 7.2, "Customer Flows in Comms Agent Assisted Billing Care"

- Section 7.3, "Best Practices for Flows"

- Section 7.4, "Merge Process in Customer MDM"

- Section 7.5, "Merge Process with Comms Agent Assisted Billing Care"

- Section 7.6, "Best Practices for Using the Merge"

- Section 7.7, "Routing Rules Configuration"

7.1 Customer Flows in Customer MDM

Upon creation of a new customer in Siebel Customer Relationship Management (Siebel CRM), the Oracle Customer MDM integration sends the customer data to the Oracle Customer Hub. The customer data then goes through a sanitation process within the Oracle Customer Hub where it is cleansed, standardized, matched, and linked to other existing customer entries and possibly enriched. The updated customer data is then returned to Siebel CRM and published to other subscribing applications.

7.2 Customer Flows in Comms Agent Assisted Billing Care

In the Comms Agent Assisted Billing Care integration, upon an update of the customer data, the customer data is published to Oracle Application Integration Architecture (Oracle AIA). If the customer data is cross-referenced in Oracle Billing and Revenue Management (Oracle BRM), then Oracle AIA applies the updates to the customer data in Oracle BRM.

When the two integrations are used together, the flow in the Comms Agent Assisted Billing Care integration that updates the customer data from Siebel CRM to Oracle BRM is performed as usual. In addition, customer data updates are published as a customer data sync from Oracle Customer Hub to Oracle BRM. This is done as a batch publish, enabling the delay introduced in the batch ensures that the updates from Oracle Customer Hub come to Oracle BRM after the updates from Siebel CRM.

Figure 7–1 shows the functional steps when updating account level customer data:

Figure 7–1 Updating Account-Level Customer Data Across MDM Customer and AIACOM Flows
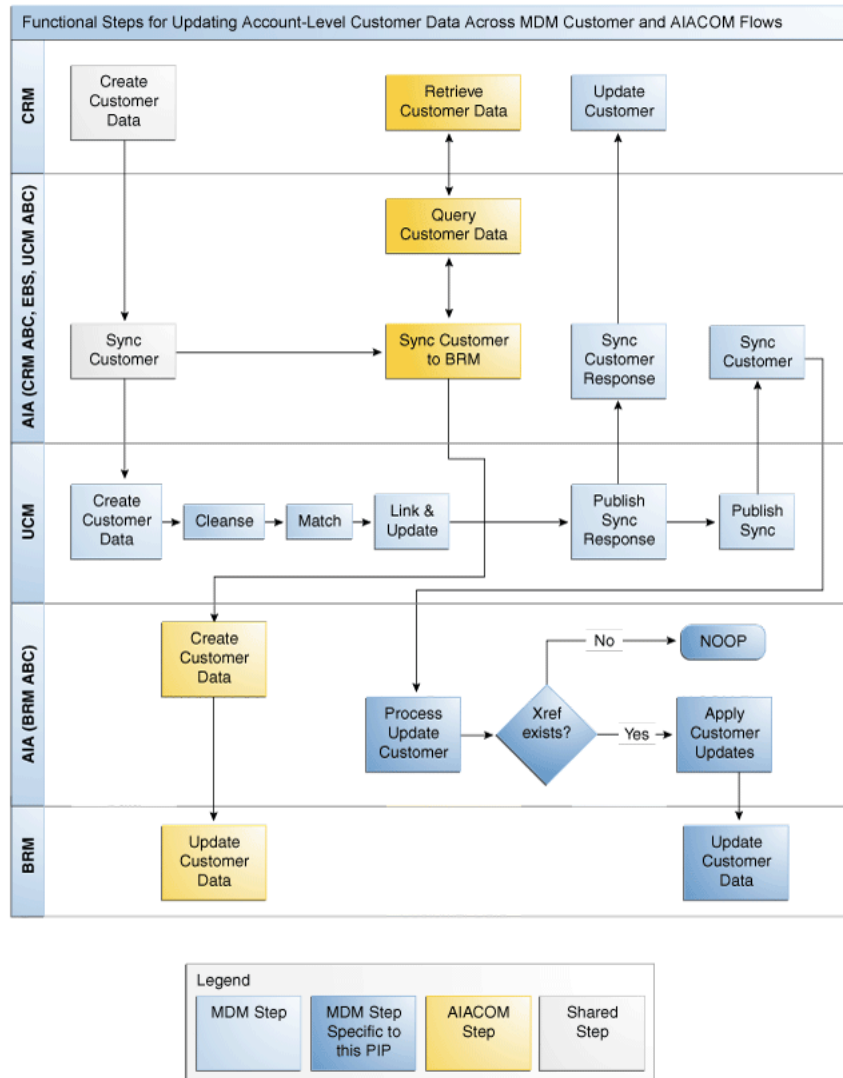
Figure 7–2 shows the functional steps when updating contact level customer data:

Figure 7–2   Updating Contact-Level Customer Data Across MDM Customer and AIACOM Flows
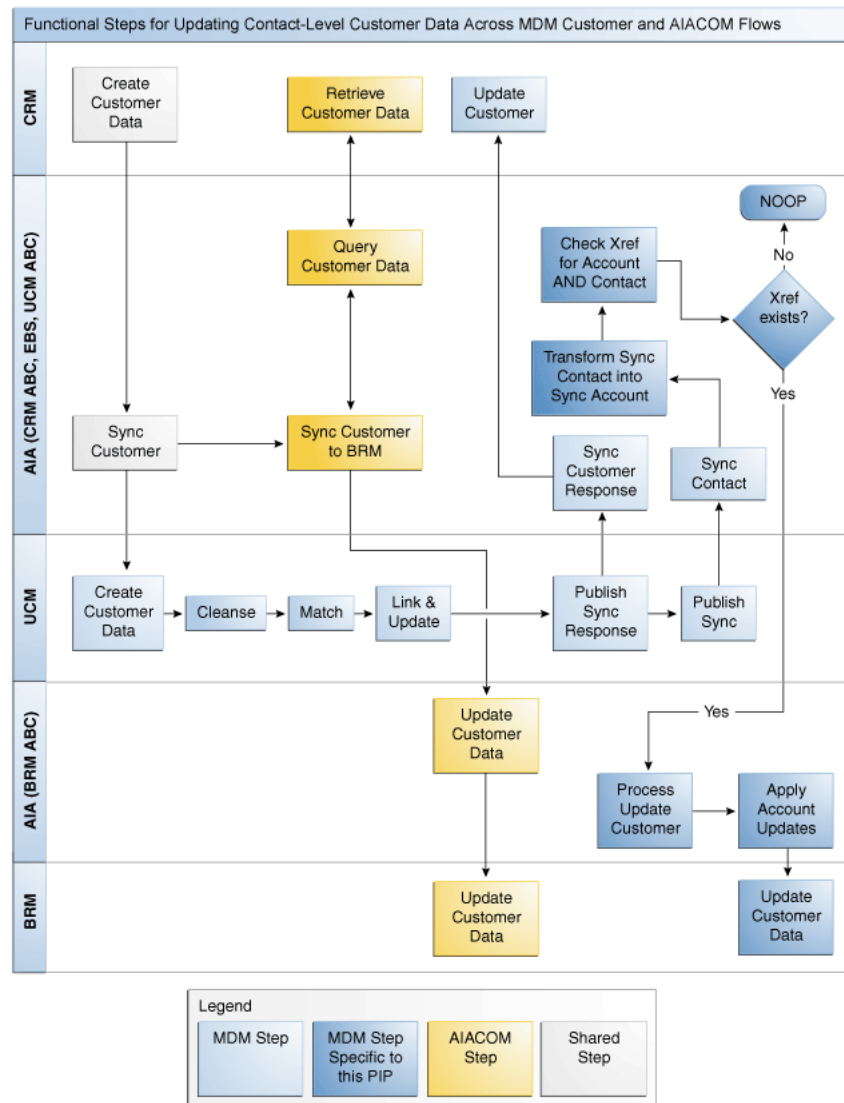
For more information, see the *Oracle Application Integration Architecture Customer Master Data Management Integration Pack Implementation Guide* and the *Oracle Application Integration Architecture Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care Implementation Guide*.

7.3 Best Practices for Flows

These are the recommendations:

- The Comms Agent Assisted Billing Care flow for updating the customer data from Siebel CRM to Oracle BRM does not change when used with the Customer MDM integration.

- The sync customer message published by Oracle Customer Hub upon updating customer data is published to Oracle BRM as well.

  The routing rules and the Oracle BRM provider check if a cross-reference for the customer data exists and, if so, it uses the data published to update the existing

customer. If the cross-reference is not found, then the data published by the Oracle Customer Hub is discarded.

- Publishing from Oracle Customer Hub to Oracle BRM is done in batch.

  How often the batch must be run depends on specific deployment conditions. It can be set from near-realtime sync (minimum delay in the batch) to a daily batch. Factors that must be considered in determining the frequency of the batch publishing include: frequency of bill cycles, performance impact, and possible requirements for having near-realtime updated customer data in the billing system.

- When interoperating with applications that do not support a merge process, if you used the merge in Oracle Customer Hub, it is recommended that you turn off the publish to the participating applications.

  For more information, see Section 7.6.1, "To Disable the Publish Process".

- To enable functional interoperability, the AIA configuration property ProcessUpdateEventsOnly must be set to *false* when both of these integrations are deployed. The default value for this property is *false*, but certain flows require that this property is set to *true*. You must change this configuration property based on the flows being used:

  - If customers are only using the Create Customer and the Update Customer flows in the MDM Customer integration, then the ProcessUpdateEventsOnly property must be *false*.

  - If customers are only using Comms Order to Cash and Comms Agent-Assisted Billing Care, then the value of the flag must be set to *true*. This is an optimization guideline, in that in the Comms flows, customer creation is enabled through the order flows. Consequently, setting the value to *true* ensures that only update events are propagated outward from the Siebel CRM customer sync requestor service.

  - However, if customers are using both Customer MDM and Comms Order to Cash or Comms Agent-Assisted Billing Care, then the property must be *false* to enable the creates from Siebel CRM to be propagated to the Oracle Customer Hub.

7.4 Merge Process in Customer MDM

In the Customer MDM integration, the merge functionality is a core function that allows a data steward to effectively determine a survivor record among several duplicate records, or to create a new record to consolidate several duplicate or similar records. For this process, Oracle Customer Hub publishes the results of the merge by providing the **survivor** record and one or many **victim** records to the source systems. It is up to the source systems to determine whether they can or should consume this message in their own records. Although some participating applications have the ability to request a merge operation, only customer merges executed within Oracle Customer Hub are supported for this release.

7.5 Merge Process with Comms Agent Assisted Billing Care

Siebel CRM has the capability and a service to merge records and re-parent the associated entities to reflect the merge performed by Oracle Customer Hub. Unfortunately, Oracle BRM does not offer a service to support this feature. Obviously merging accounts or contacts in Siebel CRM without merging the corresponding accounts or contacts in Oracle BRM is not a feasible option since it is not possible to

retrieve invoice details for victim accounts or submit change orders related to assets parented to victim accounts.

Clients can develop ad-hoc procedures and tools so that administrators can merge customer data in Siebel CRM and in Oracle BRM to be consistent with the merge data in Oracle Customer Hub. This would be a custom solution not supported by the delivered flows.

7.6 Best Practices for Using the Merge

The following are the recommendations for using the merge process:

- Account or contact merges can take place only in Oracle Customer Hub.

- Merge messages from Oracle Customer Hub are routed to all target applications registered within Oracle Customer Hub, including Oracle BRM.

- Merges are not done in Oracle BRM when receiving a merge message.

  The Oracle BRM provider checks the cross-reference for the survivor customer data and if it exists, updates the survivor customer data. This is the same behavior for a sync message on the survivor.

7.6.1 To Disable the Publish Process

When interoperating with applications that do not support a merge process, if you used the merge in Oracle Customer Hub, it is recommended that you turn off the publish to the participating applications.

Therefore, for Comms Agent Assisted Billing Care processes that interoperate with MDM for Customer processes, when a merge happens in Oracle Customer Hub, it should not be published to the CRM or BRM instances that participate in the Comms Agent Assisted Billing Care integration. If the merge is published to Siebel CRM, and the cross-references are updated, then the billing management and customer update flows break.

Perform these steps to turn off the publish after the merge:

1. Query the Oracle Customer Hub Process Merge Request workflow from **Siebel tools**.

2. Click **Revise** to revise the workflow.

3. Set the value of the process property **EnablePubSub = False**.

4. Deploy the workflow.

For more information, see the Siebel Book Shelf: **Application Development**, **Siebel Business Process Framework**, **Workflow Guide**.

7.7 Routing Rules Configuration

For Customer MDM and Comms Agent Assisted Billing Care interoperability, configuration depends on the order of deployment.

- If the Customer MDM integration is deployed first, and then the Comms Agent Assisted Billing Care integration is deployed, no routing rules changes or additions are required.

- If the Comms Agent Assisted Billing Care integration is deployed first, and then the Customer MDM integration is deployed:

1. Use the deployment plan *MDMCustomerEBSDP*.xml at $AIA_ HOME/pips/MDMCustomer/DeploymentPlans to redeploy CustomerPartyEBSV2 with overwrite flag set to true by updating the deployment plan as follows:

```
<Composite compositeName="CustomerPartyEBSV2"
compositedir="${AIA_
HOME}/pips/MDMCustomer/EBS/CustomerParty/CustomerPartyEBSV
2" revision="1.0" overwrite="true"
wlserver="pips.MDMCustomerBase" action="deploy" />
```

2. Comment out the rest of the composites and run the *MDMCustomerEBSDP*.xml deployment plan.

8

Product MDM with Communications Order to Cash

This chapter discusses the interoperable group of pre-built integrations: Oracle Product Master Data Management Integration Base Pack (Product MDM: Base), Oracle Product Master Data Management Integration Option for Siebel CRM (Product MDM: SBL CRM), Oracle Product Master Data Management Integration Option for Oracle Communications Billing and Revenue Management (Product MDM: Comms BRM) and Oracle Communications Order to Cash Integration Pack for Siebel CRM, Oracle OSM, and Oracle BRM (Comms Order to Cash). This chapter includes the routing rules that must be changed to enable interoperability.

This chapter includes the following sections:

- Section 8.1, "Product Flows in Product MDM"

- Section 8.2, "Product Flows in Comms Order to Cash"

- Section 8.3, "To Create Metadata in Oracle Product Hub"

- Section 8.4, "To Create Bundles and Promotions in Oracle Product Hub"

- Section 8.5, "Best Practices for Flows"

- Section 8.6, "Configuration Steps"

- Section 8.7, "Manual Redeployment"

8.1 Product Flows in Product MDM

In the Product MDM integration, billing products and billing discounts are defined in Oracle Product Hub (OPH). These billing products and discounts are bundled to create service bundles and promotions in OPH. These promotions are published to the target applications. The corresponding entities are created in the target applications. For example, products, discounts, bundles, and promotions are created in Siebel, but only billing products and billing discounts are created in Oracle Billing and Revenue Management (Oracle BRM).

For more information, see the *Oracle Application Integration Architecture Product Master Data Management Integration Pack Implementation Guide* and the *Oracle Application Integration Architecture Oracle Communication Order to Cash Integration Pack Implementation Guide for Siebel CRM, Oracle Order and Service Management, and Oracle Billing and Revenue Management*.

8.2 Product Flows in Comms Order to Cash

In the Comms Order to Cash integration, billing products and billing discounts are defined in (Oracle BRM) and synchronized to Siebel Customer Relationship Management (Siebel CRM). The billing products and discounts are bundled in Siebel to create service bundles and promotions. These service bundles and promotions are used in the order submission process.

8.3 To Create Metadata in Oracle Product Hub

The Item Catalog categories, attributes, and their associated value sets are manually created by the product administrator in Oracle Product Hub. These are synchronized as product classes and attributes in Siebel. These are no longer created in Siebel.

8.4 To Create Bundles and Promotions in Oracle Product Hub

Bundles (service bundles, nested service bundles, and non service bundles) and promotions are manually created by the product administrator in Oracle Product Hub (OPH) and synchronized to Siebel. Service bundles and Promotions are no longer created in Siebel CRM.

The service instance-enabled flag that identifies a simple service bundle is set in OPH and synchronized to Siebel. The value is no longer set manually in Siebel CRM.

> **Note:**   The flag value must not be changed or reset after the flag is set for a product in OPH, synchronized to the target Siebel CRM, and transactions are submitted for processing.

The association of one-time charges (suspend fee, resume fee, cancel fee, and so forth) to the corresponding bundles and account level products must be done in Siebel CRM.

The following promotion-based information is defined in OPH and synchronized to Siebel CRM. These are no longer manually defined during promotion definition in Siebel CRM:

- Promotion-based discounts.

- Promotion-based component exclusions.

- Promotion-based transaction attribute value exclusions.

The following promotion-based information is defined in Siebel CRM:

- Penalty set up for promotion cancellation or upgrade.

> **Note:**   There may be other set up steps for participating applications. See the appropriate integration implementation guide for the participating application set up steps.

8.5 Best Practices for Flows

The following scenarios are not tested and are not supported out-of-the-box for interoperability between the Product MDM integration and the Comms Order to Cash integration:

- An implementation where OPH is introduced as the Product Master when there is a single Oracle BRM instance and the product synchronization from Oracle BRM to Siebel CRM is continued to be used. That is, one set of products are defined in OPH and synchronized to Oracle BRM and Siebel CRM and another set of products are defined in the same Oracle BRM instance and are synchronized to Siebel.

- An implementation where OPH is introduced as the Product Master for Siebel CRM and one of the many Oracle BRM instances and the product synchronization from Oracle BRM to Siebel CRM is continued to be used. But the Oracle BRM instance used to synchronize products and discounts to Siebel CRM is different from the Oracle BRM instance that is mastered by OPH.

- An implementation that combines 1 and 2.

8.6 Configuration Steps

This section includes the following topics:

- Section 8.6.1, "To Create Products and Discounts in Oracle Product Hub"

- Section 8.6.2, "Routing Rules Configuration"

- Section 8.6.3, "Service Configuration Changes"

8.6.1 To Create Products and Discounts in Oracle Product Hub

Products and discounts are manually created by the product administrator in OPH. All types of billing products and discounts are supported in OPH. All the fulfillment and communications specific attributes associated with billing products and discounts are set in OPH and synchronized to Siebel CRM. These are synchronized to Oracle BRM and to Siebel CRM. Products and discounts are no longer created and published from Oracle BRM.

When OPH is used as the product master to define various entities, any product publishing mechanism that exists in other applications must be disabled. For example, in the Comms Order to Cash integration, Oracle BRM is used as the product master and publishes the products and discounts to Siebel CRM.

So in this case, when Oracle Product Hub is used as the product master, the publish mechanism in Oracle BRM must be disabled. To disable the publish, see the Oracle BRM documentation.

For more information about disabling the publish, see the Oracle BRM documentation, "Service Integration Components", "Synchronization Queue Data Manager", "Installing and Configuring the Synchronization Queue DM", "Starting and Stopping the Synchronization Queue DM".

The pricing associated with billing products is defined in Oracle Product Hub and synchronized to Siebel CRM and Oracle BRM.

The following details that are associated with products are now defined in OPH and are synchronized to Siebel CRM. These are no longer defined in Siebel CRM:

- Price overrides

- Discount overrides

- Special Rating Products

- Friends and Family lists

- Fulfillment-related information:
  - Success dependency
  - Fulfillment Item Code

The following details that are associated with discounts are now defined in OPH and are synchronized to Siebel CRM.

- Duration-based discount information.

The following flags that are associated with products, discounts, bundles, and promotions are now set in OPH and are no longer set in Siebel CRM:

- Orderable flag
- Track as Asset flag

The following information associated with billing products are enriched in Oracle BRM:

- Rate plan selectors for billing products.
- Pipeline rating for usage based billing products.
- Pricing for friends and family lists.

Discount models for billing discounts that are associated with billing discounts are enriched in Oracle BRM.

The following details are enriched in Siebel CRM:

- Matrix discounts.
- Proration plan for promotions.

8.6.2 Routing Rules Configuration

This section describes the routing rule changes that must be made to enable interoperability. There may be deployment sequence considerations.

For more information about deployment sequence considerations, see Section 1.2, "Deployment Dependencies".

For more information about creating and using routing rules, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*, "Creating Oracle Mediator Routing Rules."

> **Tip:** When OPH is used as the product master to define various entities, any product publishing mechanism that exists in other applications must be disabled.

8.6.2.1 For CommunicationPriceListResponseEBSV2

CommunicationPriceListResponseEBSV2 in the Comms Order to Cash integration has routing to CAVS because the requester does not expect a response.

However, in the Product MDM integration, the CommunicationPriceListResponseEBSV2 provides routing to the Comms2CorePriceListBridge service.

If the Comms Order to Cash integration is deployed first, and then the MDM Product integration is deployed, then CommunicationPriceListResponseEBSV2 in the MDM Product integration has to be deployed which has the following routing rule.

Table 8–1 Routing Rules for CommunicationPriceListResponseEBSV2

Operation	Filter Condition	Service Invoked	Description
SyncPriceListListResponse	No Condition	Comms2CorePriceListBridge :: execute	N/A

8.6.2.2 For CommunicationsItemCompositionResponseEBSV1

If the Comms Order to Cash integration is deployed first and the MDM Product integration is deployed second, before deploying the MDM Product integration, the following routing rule has to be removed from $AIA_HOME/pips/MDMProduct/EBS/CommunicationsItemComposition/CommunicationsItemCompositionResponseEBSV1.

Table 8–2 Routing Rules for CommunicationsItemCompositionResponseEBSV1

Operation	Filter Condition	Service Invoked	Description
SyncItemCompositionListResponse	`$in.SyncItemCompositionListResponseEBM/ebo:SyncItemCompositionListResponseEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode='PRODUCTION'and $in.SyncItemCompositionListResponseEBM/ebo:SyncItemCompositionListResponseEBM/corecom:EBMHeader/corecom:Sender/corecom:WSAddress/ns0:ReplyTo/ns0:ServiceName='SyncProductBRMComms ReqABCSImpl'`	SyncProductBRMCommsReqABCSImpl:CallbackSyncItemCompositionList	N/A

8.6.3 Service Configuration Changes

The service configuration changes are required when Communication Order To Cash Pre-Built Integration version 11.2 is used in Interoperability.

These configuration property changes are required for the ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl service. In the interop scenario on item sync, SyncItemListPIMReqABCS service waits for the response. The response from the ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl service is sent to the requester through CommunicationsPriceListResponseEBSV2.

The changes must be made in the AIAConfigurationProperties.xml for ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl service

Table 8–3 and Table 8–4 detail the property values before and after the modification. The modified property values must point to CommunicationsPriceListResponseEBSV2.

Table 8–3   Property Values Before Modification in AIAConfigurationProperties.xml for ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl service

Service	Property Values Before Modification
ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl	`<Property name="EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse.Address">http://[HTTP HOSTNAME]:[ADMIN PORT]/soa-infra/services/default/ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl/ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl</Property>`
ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl	`<Property name="EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse.PortType">{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl/V1}ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl</Property>`
ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl	`<Property name="EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse.ServiceName">{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl/V1}ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl</Property>`

Table 8–4   Property Values After Modification in AIAConfigurationProperties.xml for ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl service

Service	Property Values After Modification
ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl	`<Property name="EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse.Address">http://[HTTP HOSTNAME]:[ADMIN PORT]/soa-infra/services/default/CommunicationsPriceListResponseEBSV2/CommunicationsPriceListResponseEBSV2_ep</Property>`
ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl	`<Property name="EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse.PortType">{http://xmlns.oracle.com/ABCSImpl/Siebel/Industry/Comms/ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl/V1}ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl</Property>`
ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl	`<Property name="EBSOverride.CommunicationsPriceListResponseEBS.SyncPriceListListResponse.ServiceName">{http://xmlns.oracle.com/EnterpriseServices/Core/PriceList/V2}CommunicationsPriceListEBSResponse</Property>`

8.7 Manual Redeployment

The Product MDM integration and the Comms Order to Cash integration can be deployed in any order of sequence, but if this is a fresh environment, it is recommended that you deploy the Product MDM integration first and then the Comms Order to Cash integration.

If you do so, there are no routing rule setup required.

However, if you have deployed Comms Order to Cash integration first and then the Product MDM integration, follow the respective MDM Product section to complete the routing rule configuration

For Product MDM and Comms Order to Cash interoperability, configuration depends on the order of deployment.

- If the Product MDM integration with its BRM spoke is deployed first, and then the Comms Order to Cash integration is deployed, then no extra action is required.

- If the Comms Order to Cash integration is deployed first, and then the Product MDM integration is deployed:

  1. Use the deployment plan *MDMProductDP*.xml at $AIA_HOME/pips/MDMProduct/DeploymentPlans to redeploy CommunicationsPriceListResponseEBSV2 and CommunicationsItemCompositionResponseEBSV1 with overwrite flag set to true by updating the deployment plan as follows:

     ```
     <Composite
     compositeName="CommunicationsPriceListResponseEBSV2"
     compositedir="${AIA_
     HOME}/pips/MDMProduct/EBS/CommunicationsPriceList/Communic
     ationsPriceListResponseEBSV2" revision="1.0"
     wlserver="pips.MDMProductPIM" action="deploy"
     overwrite="true"/>
     ```

     ```
     <Composite
     compositeName="CommunicationsItemCompositionResponseEBSV1"
     compositedir="${AIA_
     HOME}/pips/MDMProduct/EBS/CommunicationsItemComposition/Co
     mmunicationsItemCompositionResponseEBSV1" revision="1.0"
     wlserver="pips.MDMProductPIM" action="deploy"
     overwrite="true"/>
     ```

  2. Comment out rest of the composites and run the *MDMProductDP*.xml deployment plan.

     You may also have to manually redeploy all the extra routing rules that you have deployed as part of the implementation of multiple Oracle BRM systems.